

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

URČOVÁNÍ POLOH ROBOTŮ TRILOBOT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

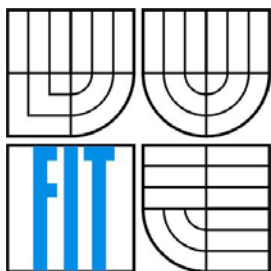
AUTHOR

TOMÁŠ LOYKA

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

URČOVÁNÍ POLOH ROBOTŮ TRILOBOT

DETERMINATION OF TRILOBOT ROBOT POSITIONS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ LOYKA

VEDOUCÍ PRÁCE
SUPERVISOR

DOC. ING. FRANTIŠEK ZBOŘIL, CSC

BRNO 2007

Zadání:

1. Proveďte přehled současného stavu dané problematiky
2. Zaměřte se na způsob lokalizace robotů v laboratoři s využitím statické kamery
3. Navrhněte program pro lokalizaci robotů a určování jejich vzájemných poloh
4. Navrhněte program pro určování vzájemné polohy robota a objektu, který má uchopit.
5. Proveďte experimenty a zhodnoťte dosažené výsledky

Licenční smlouva

Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

Abstrakt

Tato diplomová práce se zabývá technikami počítačového vidění, metodami zpracování a rozpoznávání obrazu. Smyslem je navrhnout aplikaci pro určení vzájemných poloh robotů Trilobot v laboratoři.

Klíčová slova

počítačové vidění, zpracování obrazu, digitalizace obrazu, předzpracování obrazu, segmentace obrazu, popis objektů, klasifikace objektů, prahování, detekce hran, filtrace obrazu

Abstract

This master's thesis is engaged in machine vision, methods of image processing and analysis. The reason is to create application to determine relative positions of Trilobot robots in the laboratory.

Keywords

machine vision, image processing, image digitalization, image pre-processing, image segmentation, shape representation and description, image understanding, threshold, edge detection, image filtration

Citace

Tomáš Loyka: Určování poloh robotů Trilobot, diplomová práce, Brno, FIT VUT v Brně, 2007

Určování poloh robotů Trilobot

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Doc. Ing. Františka Zbořila, CSc.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Loyka
1.5.2007

Poděkování

Děkuji Doc. Ing. Františku Zbořilovi, CSc. nejen za odborné vedení, rady a podněty, které mi během práce poskytoval, ale také za jeho vstřícnost při odborné pomoci a poskytování informačních zdrojů.

© Tomáš Loyka, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod	3
2 Vlastnosti lidského systému vidění	4
2.1 Elektromagnetické záření	4
2.2 Lidské oko	5
3 Snímání a digitalizace obrazu	7
3.1 Snímání obrazu	7
3.2 Digitalizace obrazu	8
3.2.1 Vzorkování obrazu	8
3.2.2 Kvantování obrazu	9
3.2.3 Popis barev v obraze	9
3.3 Vlastnosti digitálního obrazu	11
4 Předzpracování obrazu	13
4.1 Jasové transformace	13
4.2 Geometrické transformace	15
4.3 Filtrace a ostření	16
5 Segmentace obrazu	18
5.1 Prahování obrazu	19
5.2 Detekce hran	21
5.3 Srovnávání se vzorem	23
6 Popis objektů	24
6.1 Identifikace oblastí	25
6.2 Popis objektů založený na popisu hranic	25
6.3 Popis objektů založený na popisu plochy	27
7 Klasifikace objektů	31
7.1 Příznakové rozpoznávání	31
7.2 Syntaktické rozpoznávání	32
8 Návrh programu	33
8.1 Robot „Trilobot Research Robot“	33
8.2 Příprava robotů a laboratoře	34
8.3 Digitalizace obrazu	37
8.4 Předzpracování obrazu	38
8.4.1 Tvorba histogramu	38
8.4.2 Prahování obrazu	39

8.4.3	Zvýraznění barev v obraze.....	42
8.4.4	Odstranění šumu, ostření obrazu.....	44
8.4.5	Detekce hran	48
8.5	Segmentace obrazu, klasifikace objektů	52
8.5.1	Vyhledání robotů	52
8.5.2	Vyhledání identifikačních značek.....	54
8.5.3	Vyhledání míčů.....	55
8.6	Vztahy mezi objekty v obraze.....	57
9	Závěr	58
	Literatura	59
	Seznam příloh.....	60

1 Úvod

Téma této práce zní: Určování poloh robotů Trilobot. Co si pod tímto názvem představit? Určovat polohu objektů můžeme v dnešní době mnoha různými způsoby, zde se však budeme zabývat určováním polohy několika robotů a dalších objektů v laboratoři pomocí obrazové informace, tedy rozpoznáváním obrazu.

Problematika rozpoznávání obrazu je částí disciplíny zvané počítačové vidění. Počítačové vidění je disciplína, která se snaží technickými prostředky napodobit lidské vidění. Pro počítačové vidění je typická snaha porozumět obecné trojrozměrné scéně, např. takové, jakou můžete vidět při pohledu z okna. Postupy počítačového vidění jsou značně složité, s těžištěm v interpretaci obrazových dat, která jsou nejčastěji reprezentována symbolicky. Jádrem pokročilejších postupů jsou znalostní systémy a techniky umělé inteligence. Této části počítačového vidění se říká vyšší úroveň, první částí je nižší úroveň.

Cílem nižší úrovně je analyzovat vstupní dvojrozměrná obrazová data číselného charakteru a najít kvalitativní symbolickou informaci potřebnou pro vyšší úroveň. Pro nižší úroveň se také používá název zpracování obrazu počítačem.

Předmětem zpracování a případné rozpoznání obrazu je obrazová informace o reálném světě, která do počítače vstupuje televizní kamerou, fotoaparátem, skenerem. Počítačové vidění řeší úlohu vytvoření explicitního popisu fyzikálních objektů v obraze.

Vlastní průběh zpracování a rozpoznávání obrazu reálného světa obvykle rozdělujeme do několika základních kroků. Toto rozdělení však není zcela jednoznačné, záleží již na dané aplikaci, zda budou provedeny všechny kroky tak, jak jsou zde uvedeny.

Jednotlivé kroky jsou:

- Snímání a digitalizace obrazu
- Předzpracování
- Segmentace obrazu
- Popis objektů
- Klasifikace

V první části této práce se budu zabývat teoretickými poznatky týkajícími se počítačového a lidského způsobu vidění, různými metodami a technikami zpracování a interpretace obrazu. Druhá část práce poté pojednává o praktických poznatcích a přístupech, které jsem získal při řešení tohoto projektu.

Tímto tématem jsem se zabýval již v rámci mých předešlých projektů, kdy jsem řešil úlohu najít polohu jediného robota v laboratoři ve vymezeném prostoru.

2 Vlastnosti lidského systému vidění

Dříve, než se budeme zabývat počítačovým viděním a zpracováním obrazu je vhodné, abychom se, alespoň v základu, seznámili s lidským systémem vidění, porozuměli tomu, co to vlastně obraz je, jak vzniká, co to je barva, jas, a jakým způsobem je člověk schopen obraz vnímat.

2.1 Elektromagnetické záření

Elektromagnetické záření je vlnění, které vzniká oscilací elektricky nabitých materiálů. Pouze malou část tohoto záření je člověk schopen vnímat, tato část se nazývá viditelné světlo. Elektromagnetické spektrum zahrnuje všechny známé druhy záření: od radiových vln, přes mikrovlny, viditelné světlo, až po rentgenové a gama záření. Elektromagnetické záření může reagovat s látkami různým způsobem, v závislosti na vlnové délce. Obrazy pořízené při různých vlnových délkách tedy mohou mít odlišné vlastnosti a poskytovat o objektech rozdílné informace.

Viditelná část spektra se nachází v oblasti vlnových délek přibližně 380 až 720 nm. Uvnitř této oblasti vnímáme záření s určitou vlnovou délkou jako barvu. Světlo o vlnové délce 720 nm se jeví jako červené, o vlnové délce 550 nm jako zelené a o vlnové délce 380 nm jako fialové.

Rentgenové a gama záření nachází uplatnění v lékařství. Tato záření s kratší vlnovou délkou než viditelné světlo mají totiž energii a tudíž větší schopnost pronikat látkou než viditelné světlo. Toho se využívá při zobrazení vnitřní struktury těla (Rentgenové záření), nebo při zjišťování pohybu látky v těle (přidání příměsi která vyzařuje gama záření do vstříknuté látky).

Infračervené záření, jehož vlnová délka je delší než světlo, má také význam při zobrazování. Každý objekt, který je zahřátý na nějakou teplotu, totiž vyzařuje infračervené světlo, a tudíž jej lze podle něj zobrazit a lokalizovat i ve tmě.

Světelný zdroj jako například žárovka nebo Slunce vyzařuje paprsky všech frekvencí v daném pásmu, které se pak skládají ve výsledné bílé světlo. Takové světlo (všech barev) se nazývá achromatické. Dopadne-li bílé světlo na objekt, jsou některé frekvence od objektu odrazeny a některé pohlceny. Frekvence, které se vyskytují v odraženém světle vnímáme jako barvu objektu. Převládají-li tedy v odraženém světle nízké frekvence, vnímáme objekt jako červený.

Pro popis charakteristik světla jsou důležité kromě frekvence ještě další hodnoty. Jsou to:

Jas – určuje intenzitu světla

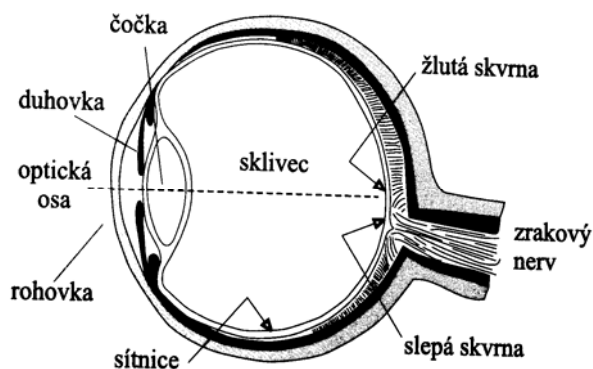
Sytost – udává čistotu barvy světla. Je tím vyšší, čím užší je frekvenční spektrum.

Světlost – určuje velikost achromatické složky ve světle s určitou dominantní frekvencí

Toto jsou základní charakteristiky, které můžeme použít pro popis vlastností, které vnímáme u zdroje světla.

2.2 Lidské oko

Lidské oko má přibližně tvar koule o průměru 2 cm. V přední části oka je rohovka, která zajišťuje ochranu oka. Za rohovkou je duhovka, která se může otevírat nebo zavírat, a tím reguluje množství světla, které vstupuje do vnitřní části oka. Za duhovkou je čočka, která zajišťuje zaostření vstupujících světelných paprsků. Čočka pohlcuje asi 8% dopadajícího světla. Čočka je připojena ke svalům, které mohou měnit její geometrii a tím i její ohniskovou vzdálenost. Zaostřený obraz se poté promítá na sítnici. Je to tenká fotocitlivá vrstva na vnitřním povrchu oka.



obr. 2.1: Lidské oko

Sítnice obsahuje dva druhy fotocitlivých receptorů: tyčinky a čípky. Tyčinky zajišťují noční vidění. Jsou totiž asi 10x citlivější než čípky a reagují i na velmi nízkou úroveň osvětlení. Čípky zajišťují barevné vidění a reagují na vyšší hodnoty světla. Každý čípek obsahuje jeden ze tří druhů fotopigmentů, který určuje na jakou vlnovou délku světla je čípek citlivý. Fotopigmenty se označují jako červený, zelený a modrý. Fotopigmenty nejsou na sítnici zastoupeny ve stejném poměru. V oku je přibližně 64% čípků s červeným fotopigmentem, 32% se zeleným a jen 2% s modrým fotopigmentem. Z toho také vyplývá odlišná citlivost oka na různé barevné frekvence.

Tyčinky a čípky nejsou také rozmístěny v oku pravidelně, největší koncentrace čípků je v oblasti optické osy oka, jinde je naopak převaha tyčinek. V oblasti tzv. slepé skvrny se pak nevyskytují ani tyčinky ani čípky, je to místo, ve kterém se všechny spoje od fotoreceptorů sbíhají do zrakového nervu. Oko obsahuje přibližně 130 milionů fotoreceptorů, naproti tomu zrakový nerv má asi jen 1 milion vláken. Z toho tedy vyplývá, že značná část zpracování obrazové informace probíhá přímo na sítnici: vždy několik tyčinek je připojeno k jednomu zrakovému nervu, tyčinky tedy poskytují obraz z nižším rozlišením. Důsledkem toho je, že ostře vidíme jen ve středu zorného pole, kde je převaha čípků.

Ve zrakovém nervu, který předává signály od fotoreceptorů do mozku, dochází k rekombinaci barevné informace. Ze tří původních kanálů r, g, b vznikají tři odlišné kanály. Jeden nese informaci o poměru barev červená-zelená, druhý o poměru barev žlutá-modrá. Tyto kanály poskytují informaci o

barvě. Třetí kanál nese informaci zelená+červená, čímž indikuje jas. V důsledku toho nejsou lidé schopni vnímat některé barevné kombinace (modro-žlutá, červeno-zelená).

Co se týče jasu, tak lidské oko je schopno se adaptovat na obrovský rozsah úrovní světla. Podstatně více než jakýkoliv elektronický systém. Horní a dolní meze intenzity se liší násobkem 10^{10} . Není však samozřejmě schopno vnímat takový obrovský rozsah současně, dochází k adaptaci.

3 Snímání a digitalizace obrazu

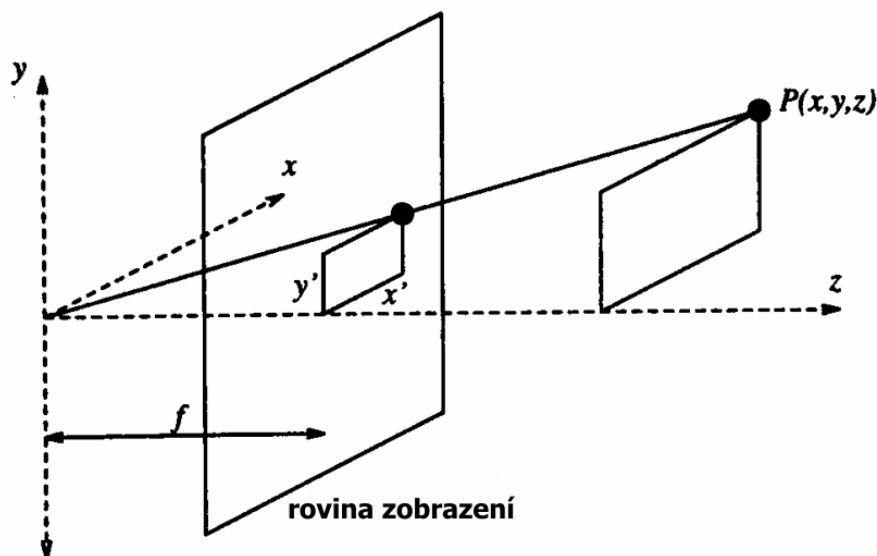
Základním požadavkem pro zpracování a rozpoznání obrazu je vlastní získání obrazu reálného světa a jeho převod do digitální podoby vhodné pro uložení a další zpracování v počítači.

3.1 Snímání obrazu

Snímání obrazu je převod optické veličiny na elektrický signál, který je spojitý v čase i v úrovni. Vstupní informací pro snímání nemusí být vždy jen jas z kamery či scanneru, ale mohou to být i jiné veličiny, jako jsou tepelné záření, rentgenové záření či ultrazvuk. Snímání dále může probíhat buď v jednom nebo v několika spektrálních pásmech. Pro barevné snímání postačí složky tři: červená, zelená a modrá.

Na výsledný sejmutý obraz má vliv mnoho různých faktorů. Může to být například osvětlení snímaného objektu, jeho vlastnosti, odrazivost, a další.

Snímáním získáme 2D obraz dané scény, který je perspektivní projekcí 3D prostředí, které snímáme. Geometrie perspektivního zobrazení je ukázána na obrázku.



obr. 3.1: Projekce perspektivního zobrazení

Nechť x , y a z jsou souřadnice bodu P ve 3D scéně a f je ohnisková vzdálenost čočky. Potom bod P má po perspektivním zobrazení souřadnice (x', y') v rovině zobrazení, kde:

$$x' = \frac{xf}{z}, \quad y' = \frac{yf}{z} \quad (3.1)$$

Tento model je velice realistický, a vlastně odpovídá obrazu, který vzniká na sítnici lidského oka. Mnoho informací se však ztratí, a následná rekonstrukce 3D scény z 2D nasnímaného obrazu je

velice obtížná, a možná jen za předpokladu, že o snímané scéně máme dostatečný počet informací. Při snímání a rekonstrukci 3D scény se však většinou používají jiné metody, které jsou založeny na vyšším počtu snímacích zařízení a snímání scény z několika pohledů. Řešení úloh, které mají 2D povahu je tudíž mnohem jednodušší, a ve velké většině případů i dostatečné.

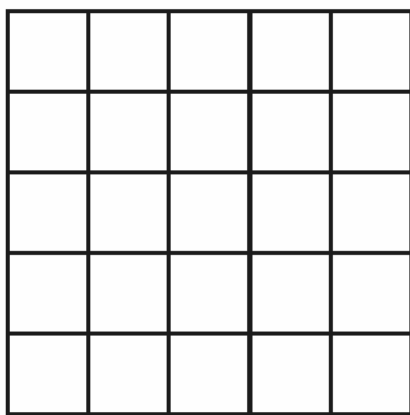
3.2 Digitalizace obrazu

Druhým krokem při získávání obrazu vhodného pro další zpracování v počítači je převod spojitého analogového signálu odpovídajícího monochromatickému signálu na signál digitální – digitalizace.

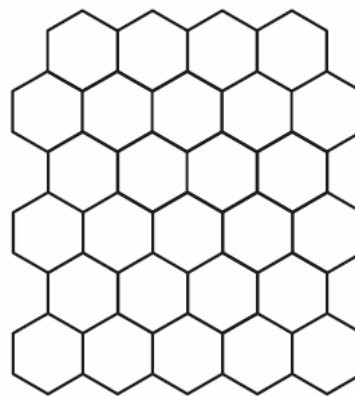
Digitální obraz je ekvivalentem spojitě obrazové funkce $f(i,j)$, kde i a j jsou souřadnice v obraze. Je získán **vzorkováním** obrazu do matice $M \times N$ bodů a **kvantováním** do K úrovní jasu. Jednomu prvku matice se říká obrazový element (picture element – pixel). Z hlediska zpracování obrazu jde o dále nedělitelnou jednotku.

3.2.1 Vzorkování obrazu

Při vzorkování obrazu převádíme spojitou obrazovou funkci $f(x,y)$ do diskrétní mřížky obrazových bodů. Důležitou součástí vzorkování je tedy volba vzorkovací mřížky. Nejčastěji používanými mřížkami je mřížka čtvercová a hexagonální. Jejich struktura je naznačena na obrázku.



obr. 3.2: čtvercová mřížka



hexagonální mřížka

Čtvercová mřížka je nejobvyklejší, vychází z konstrukce většiny snímacích prvků, a ze struktury paměti a je tudíž velice snadno realizovatelná. Její nevýhody se však projeví při měření vzdálenosti mezi jednotlivými body, a také při zjišťování spojitosti objektů. Hexagonální mřížka tyto problémy řeší, avšak je obtížněji realizovatelná a pro některé operace i méně vhodná. Dále budeme tedy uvažovat jen čtvercovou vzorkovací mřížku.

Vzorkování se dále řídí známým Shannonovým vzorkovacím teorémem. Z to plyne, že nejmenší detail v digitálním obraze musí být minimálně dvojnásobkem vzorkovacího intervalu. Volba vhodného rozlišení je tedy jedním z nejzásadnějších kroků digitalizace. Při nízkém rozlišení budeme

ztrácet informace o detailech v obraze, které bychom mohli dále potřebovat, a naopak při vysokém rozlišení nám bude stoupat výpočetní náročnost a tedy i spotřebovaný čas na další zpracování obrazu.

Velikost obrazu se většinou udává v pixelech, rozlišení obrazu je obvykle udáváno v jednotkách body/palec (dots per inch – DPI).

3.2.2 Kvantování obrazu

Kvantování probíhá v oboru hodnot obrazové funkce (jasu), který se rozdělí na intervaly, jimž je pak přidělena jediná, zástupná hodnota. Počet intervalů volíme tak, aby byl dostatečný pro lidské vnímání a rozlišení jemných detailů v obraze. Podle způsobu rozdělení kvantované veličiny hovoříme o kvantování uniformním a neuniformním.

Uniformní kvantování používá konstantní délku intervalu, zatímco neuniformní kvantování používá proměnnou délku intervalu. Neuniformní kvantování se používá méně často. Důvodem je značně jednodušší realizovatelnost uniformního kvantování.

U obrazů, který byly kvantovány nedostatečným počtem úrovní, je hlavním problémem výskyt falešných obrysů. Tento efekt nastává pokud kvantovaný počet úrovní jasu je nižší než dokáže člověk lehce rozlišit. Toto číslo je závislé na mnoha okolnostech, například průměrná světlost, ale od počtu úrovní 100 a více už k tomuto jevu nedochází.

3.2.3 Popis barev v obraze

Barva obrazu je velmi důležitým nositelem informace. Pro určení přesného odstínu barvy je vytvořeno několik různých barevných modelů. Ty se liší podle toho, které složky jsou při popisu barvy použity.

3.2.3.1 Model RGB

Tento barevný model je využíván v zobrazovacích zařízeních (monitor, TV), kde jeden viditelný bod (pixel) tvoří tři velmi blízko umístěné body barevných složek Red, Green, Blue (červená, zelená, modrá). Jejich sloučením vznikají další barvy různých odstínů.

Tento model je aditivní, to znamená, že pro hodnoty RGB (0,0,0) je výsledná barva černá. Se zvyšující hodnotou se jas zvyšuje, až po maximální hodnotu (1,1,1) kdy je výsledná barva bílá. Pokud jsou hodnoty jednotlivých složek stejné, jedná se o odstín šedé, pokud jsou hodnoty složek rozdílné, dostáváme všechny barvy spektra.

V počítačových aplikacích se rozsah jednotlivých hodnot určuje v celých číslech, většinou v rozmezí 0-255. Toto rozmezí se nazývá také barevná hloubka.

Barevná hloubka 8 bitů je tvořena tabulkou 256 definovaných barev. Dále se používá barevná hloubka 24 bitů, tedy 256 hodnot pro každou složku RGB. Tato barevná hloubka se nazývá také

TrueColor, protože je možno zobrazit již všechny odstíny barev, které je možné vidět ve skutečném světě.

Pokud bychom chtěli převést různobarevný obraz v RGB modelu na šedotónový, nemůžeme jeho barvy nahradit odstíny šedi získanými prostým průměrem ze tří základních barev. Lidské oko totiž vnímá různým způsobem intenzitu jednotlivých barevných složek (nejcitlivější je na zelenožlutou), takže pro výpočet jasu se používá vztah:

$$I = 0.299R + 0.587G + 0.114B \quad (3.2)$$

3.2.3.2 Modely CMY a CMYK

Tento barevný model je využíván pro tisk a při výrobě fotografií. Na rozdíl od předchozího modelu se jedná o subtraktivní systém. Barvy se neskládají, ale odečítají od původní bílé. Se zvyšující hodnotou dané složky je tedy celkový jas nižší, složením všech barev vznikne černá.

Vztah mezi RGB a CMY (Cyan, Megenta, Yellow – azurová, purpurová, žlutá):

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.3)$$

Barva (1,0,0) v RGB je tedy ekvivalentní barvě (0,1,1) v CMY.

Model CMYK doplňuje původní složky CMY ještě o složku K (blacK - černá). Tento model se používá často v tiskárnách, jinak by bylo nutné nanést všechny tři složky CMY a výsledná barva by stejně nebyla černá, ale spíše šedá.

3.2.3.3 Model HSI

Na rozdíl od předchozích modelů nejsou jeho složky tvořeny jednotlivými základními barvami, ale jejich třemi vlastnosti: Hue, Saturation, Intensity (barevný tón, sytost, jas).

Barevný tón je udáván jako úhel v rozmezí 0-360°. Barvy tvoří uzavřený kruh a jsou lineární. Lze tedy zadat libovolný zlomek úhlu pro určení požadované barvy.

Sytost určuje množství přidané bílé složky.

Jas určuje, kolik světla barva odrazí, tedy jak zářivá bude.

Tento model má výhodu v tom, že zohledňuje lidské vnímání barev, umožňuje postupně měnit barevné charakteristiky při zachování ostatních typických vlastností barvy. To je příjemné pro uživatele, kteří chtějí definovat barvy přirozenými pojmy. Používá se tedy v některých algoritmech při digitálním zpracování obrazu.

3.2.3.4 Modely YUV, YIQ

Tyto barevné modely se při zpracování obrazu nepoužívají. Jejich využití je v televizním vysílání, model YUV se používá pro přenos televizních signálů v normě PAL, model YIQ pro normu NTSC.

Jejich společným rysem je oddělení jasové složky od barevných, což umožňuje využít jeden jasový signál jak pro barevné, tak pro černobílé televizory. Složka Y zde představuje jas nebo šedost, a U a V (I a Q) jsou barevné složky.

3.2.3.5 Šedá škála

Často vystačíme s obrazem v šedé škále. To znamená rozmezí bílá, stupně šedé, černá. Většinou se používá 256 odstínů šedi, tedy barevná hloubka 8 bitů.

Pro převod barevného obrazu v modelu RGB do obrazu šedé škály používáme vzorec (3.2).

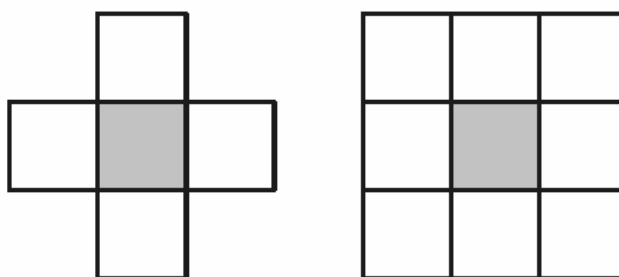
3.3 Vlastnosti digitálního obrazu

V digitálním obraze můžeme popsat mnoho vlastností, jak metrických, tak topologických, z nichž některé chápeme odlišně, než jsme zvyklí ze spojitých obrazů.

Příkladem je vzdálenost mezi dvěma body obrazu (i,j) a (h,k) . Obecně je tato vzdálenost rovna Euklidovské vzdálenosti DE, definované podle vztahu:

$$D_E = \sqrt{(i-h)^2 + (j-k)^2} \quad (3.4)$$

Pokud je ale prostor diskretizován, tak definujeme nejdříve sousedství bodu a pak vzdálenost bodů v obraze za předpokladu daného sousedství. Při práci se čtvercovou mřížkou můžeme mluvit buď o 4-sousedství, nebo o 8-sousedství:



obr.3.3:

4-sousedství

8-sousedství

Vzdálenost dvou bodů je pak dána minimálním počtem kroků v digitální mřížce, které jsou potřeba k přesunutí se z jednoho bodu do druhého. Pokud máme na mysli 4-sousedství, znamená to, že jsou povoleny pouze vodorovné a svislé tahy. Tato vzdálenost bývá také nazývána „městská“, kvůli podobnosti s dvěma body ve městě a pravoúhlou sítí ulic a bloků domů. Tato vzdálenost se udává:

$$D_4 = |i-h| + |j-k| \quad (3.5)$$

Pokud uvažujeme o 8-sousedství, znamená to, že jsou povoleny i diagonální tahy. Tato vzdálenost bývá někdy nazývána „šachovnicová“. Je totiž rovna počtu tahů, kolik bychom museli udělat králem na šachovnici při pohybu z jednoho pole do druhého.

$$D_8 = \max\{|i - h|, |j - k|\} \quad (3.6)$$

Mezi další vlastnosti, kterými můžeme obraz popsat, patří histogram obrazu. Histogram ukazuje četnost výskytu jasů v obraze a je většinou jedinou informací o obrazu kterou můžeme snadno získat. Podrobněji o tom, jak histogram vypadá, co ukazuje a jak se tvoří pojednává kapitola 8.4.1.

Další vlastnosti digitálního obrazu již není tak snadné popsat a analyzovat, protože jsou založeny na lidském subjektivním vnímání a pocitech. Patří sem kontrast, jasnost a barva obrazu. Lidské vnímání také ovlivňuje množství různých optických iluzí, týkajících se jak barvy, tak i například jasu, kontrastu, nebo velikosti objektů.

Mezi poslední charakteristiku digitálního obrazu můžeme zařadit výskyt šumu. Šum je náhodná chyba v obraze, která vzniká při snímání, případně přenosu nebo zpracování a může nebo nemusí záviset na obsahu obrazu. Šum rozlišujeme podle jeho pravděpodobnostních charakteristik na několik druhů. Ideální šum, který se také nazývá bílý šum, má konstantní spektrum, což znamená, že jeho intenzita se nesnižuje se zvyšující se frekvencí. Bílý šum se často používá jako nejhorší případ degradace obrazu. Mezi další druhy šumu, které známe, patří ještě Gaussův šum a impulsní šum, nazývaný také někdy „pepř a sůl“, protože se projevuje v jednotlivých bodech obrazu, které potom mají výrazně odlišnou charakteristiku od bodů ve svém okolí (jsou typicky černé nebo bílé). O algoritmech používaných k odstranění jednotlivých druhů šumu podrobněji v kapitole 8.4.4.

4 Předzpracování obrazu

Po úspěšném získání obrazu a jeho digitalizaci máme k dispozici digitální obraz zkoumané scény. Tento obraz však může být zkreslen díky způsobu snímání nebo díky nevhodným podmínkám při jeho získání. Pokud je znám charakter zkreslení, je možné tuto chybu opravit pomocí korekcí, které jsou jednou z metod předzpracování obrazu. Existuje také však velké množství dalších metod, které usnadňují analýzu obrazu, identifikaci objektů, nebo jen zvýrazňují důležité rysy obrazu pro snazší pozorování člověkem.

Předzpracování je společný název pro operace na nízké úrovni abstrakce, kdy jsou vstupem i výstupem matice, které představují digitální obrazovou funkci.

Je důležité si uvědomit, že v průběhu předzpracování nezískáme žádnou novou informaci. Můžeme jen potlačit informaci, která není důležitá pro další zpracování, nebo naopak zvýraznit informaci, která je pro další zpracování důležitá. Metody předzpracování využívají značné nadbytečnosti údajů v obraze, po provedení transformace se hodnota informace v obraze obsažené většinou sníží.

Základní rozdělení metod předzpracování obrazu je:

- jasové transformace
- geometrické transformace
- filtrace a ostření

4.1 Jasové transformace

Jasové transformace modifikují jas bodu obrazu. Lze je rozdělit do dvou skupin. Jsou to **jasové korekce** a **modifikace jasové stupnice**. U jasových korekcí závisí jas v bodě výstupního obrazu pouze na jasu odpovídajícího bodu ve vstupním obraze (se stejnými souřadnicemi). Modifikace jasové stupnice transformuje určitou hodnotu jasu ve vstupním obraze na jinou výstupní hodnotu, a to bez ohledu na pozici v obraze.

První skupinou jsou jasové korekce. V ideálním případě by snímací a digitalizační zařízení mělo mít ve všech místech stejnou citlivost bez ohledu na umístění bodu v obraze. Tento požadavek však v praxi není často splněn. Světlo procházející dál od optické osy je více zeslabováno, také světlocitlivá vrstva snímacího senzoru nemusí mít ve všech místech stejnou citlivost. Zdrojem odchylek je i nerovnoměrné osvětlení snímané scény.

Jsou-li uvedené odchylky systematické, lze je potlačit jasovými korekcemi. Multiplikativní koeficient $e(i,j)$ popisuje odchylku každého bodu obrazu od ideální převodní charakteristiky. Pro každý bod původního (neporušeného) obrazu $g(i,j)$ získáme na výstupu zkreslený bod $f(i,j)$:

$$f(i, j) = e(i, j)g(i, j) \quad (4.1)$$

K určení chybového koeficientu $e(i,j)$ stačí při stálých snímacích podmínkách sejmut obraz $g(i,j)$ o známém průběhu jasové funkce. Nejjednodušší způsob je sejmut obraz o konstantním jasu c , který po digitalizaci označíme $f_c(i,j)$. Potom můžeme systematické chyby v digitalizovaném obraze jednoduše korigovat podle vztahu:

$$g(i,j) = \frac{f(i,j)}{e(i,j)} = \frac{c \cdot f(i,j)}{f_c(i,j)} \quad (4.2)$$

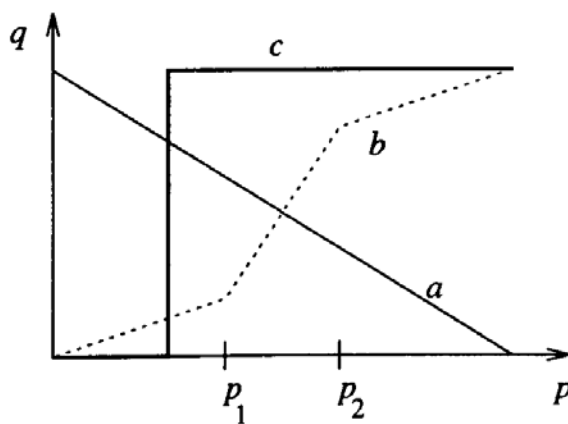
Tato metoda může být použita jen pokud jsou snímací podmínky stálé. Pokud chceme korigovat tuto chybu při měnících se snímacích podmínkách, musíme čas od času kalibrovat chybovou matici.

Také si při užití této metody musíme dát pozor, abychom při korekci nepřekročili maximální nebo minimální hodnoty jasové stupnice. Pokud snímáme referenční obraz, je tedy vhodné, aby jeho hodnota jasu byla přesně uprostřed stupnice.

Druhou skupinou je modifikace jasové stupnice. Tato transformace nezávisí na poloze bodu v obraze. Transformace T výchozí hodnoty jasu p na novou hodnotu jasu q je dána vztahem

$$q = T(p) \quad (4.3)$$

Obvyklé transformace stupnice jasu jsou vyznačeny na obrázku 4.1. Přímka a označuje negativní transformaci, po částech lineární průběh funkce b zvyšuje kontrast mezi hodnotami jasu p_1 a p_2 . Funkce c se nazývá prahování a jeho výsledkem je černo-bílý obrázek.



obr.4.1: obvyklé transformace stupnice jasu: a) negativ, b) zvýšení kontrastu
c) prahování

Digitalizovaný obraz má velmi omezené množství úrovní jasu, tudíž transformace jasové stupnice jsou velice snadno realizovatelné. Je vytvořena vyhledávací tabulka (look-up table) o tolika paměťových místech, kolik je jasových úrovní. Nové hodnoty jasu, které jsou výsledkem transformace, jsou pak přímo obsahem paměti. Vyhledávací tabulka může být zařazena do cesty obrazovému signálu, a tudíž je možné pozorovat výsledek transformace v reálném čase.

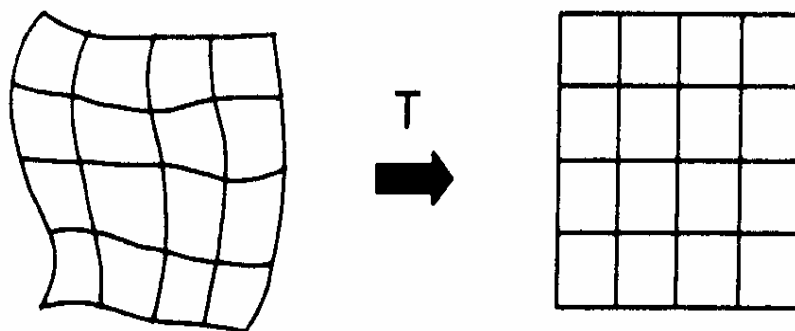
Stejný princip může být použit i pro barevné obrazy, kdy jsou vytvořeny tři vyhledávací tabulky pro jednotlivé barevné složky R, G, B.

Hodnoty jasové transformace pro zvýšení kontrastu jsou často nalezeny automaticky, podle analýzy histogramu.

Transformace jasové stupnice se používají hlavně pokud je obraz zpracováván za účelem pozdější analýzy člověkem. Po zvýšení kontrastu je obraz mnohem lépe interpretovatelný. Také se často používá transformace do falešných barev, kdy jsou určité hodnoty jasu nahrazeny barvami. Lidské oko je totiž mnohem citlivější ke změně barvy než ke změně jasu, takže člověk pak rozpozná více detailů v obraze.

4.2 Geometrické transformace

Geometrické transformace najdou uplatnění, pokud chceme provést korekci geometrického zkreslení, které bylo způsobeno při snímání obrazu, například při zpracování dokumentů OCR programem, nebo pokud chceme provést posunutí, rotaci, nebo změnu měřítka obrazu.



obr. 4.2: geometrická transformace v rovině

Geometrická transformace plošného obrazu je vektorová funkce T , která transformuje bod (x, y) v rovině na novou pozici (x', y') . Transformaci plošného obrazu ilustruje obrázek 4.2. T je definována vztahy:

$$x' = T_x(x, y) \quad y' = T_y(x, y) \quad (4.4)$$

Transformační vztahy T_x a T_y mohou být známy buď předem, jako je tomu například v případě rotace, posunu, zvětšení, nebo je možné hledat transformační vztah na základě znalosti původního i transformovaného obrazu. Při hledání vztahu se obvykle využívá několika bodů, které si odpovídají a lze je snadno najít v obou obrazech.

Pro základní geometrické transformace jsou dány vztahy:

Rotace o úhel Φ :

$$\begin{aligned} x' &= x \cos \Phi + y \sin \Phi \\ y' &= -x \sin \Phi + y \cos \Phi \end{aligned} \quad (4.5)$$

Změna měřítka a na ose x a b na ose y :

$$\begin{aligned}x' &= ax \\ y' &= by\end{aligned}\tag{4.6}$$

Zkosení o úhel Φ :

$$\begin{aligned}x' &= x + y \tan \Phi \\ y' &= y\end{aligned}\tag{4.7}$$

Geometrická transformace se skládá ze dvou kroků. První je **plošná transformace** bodu, která najde vstupnímu bodu s diskrétními souřadnicemi odpovídající výstupní bod. Souřadnice výstupního bodu musí být počítány jako spojité, protože jeho poloha po transformaci obvykle nesouhlasí s celočíselnou mřížkou. Druhým krokem je **nalezení úrovně jasu** v konkrétním bodě mřížky, který se obvykle interpoluje z několika bodů v okolí, které po provedení transformace nemají celočíselné souřadnice. Toto okolí bývá většinou malé, vzhledem k výpočetní náročnosti. Tři nejpoužívanější metody jsou metoda nejbližšího souseda, která je nejjednodušší a nejrychlejší – hodnota bodu mřížky se určí podle nejbližšího výstupního bodu, dále potom lineární interpolace – hodnota bodu mřížky se určí průměrem 4 sousedních výstupních bodů – tato metoda způsobí slabé zjemnění a rozmazání obrazu a poslední a nejpřesnější metodou je bi-kubická interpolace, která využívá pro určení hodnoty bodu 16 nejbližších výstupních bodů.

4.3 Filtrace a ostření

Tyto metody využívají pro výpočet jasu bodu ve výstupním obraze jen lokálního okolí odpovídajícího bodu ve vstupním obraze, proto se jim také někdy říká lokální metody předzpracování obrazu. Podle cíle zpracování můžeme tyto metody rozdělit do dvou velkých skupin: vyhlazování a gradientní operace (detekce hran).

Vyhlazování obrazu vede k potlačení vyšších frekvencí obrazové funkce. Žádaným výsledkem vyhlazování je tedy potlačení náhodného šumu a drobných odchylek. Bohužel vyhlazování také potlačuje ostatní náhlé změny jasové funkce, dochází tedy k oslabení ostrých čar a hran.

Gradientní operace a ostření obrazu naopak vedou ke zdůraznění vyšších frekvencí. Výsledkem je tedy zvýraznění těch částí obrazu, kde se jasová funkce náhle mění, dochází tedy ke zvýraznění hran. Nežádoucím výsledkem je současné zvýraznění šumu v obraze.

Vyhlazovací a gradientní operace mají tedy protichůdné cíle. Řešením tohoto rozporu jsou některé algoritmy, které oba postupy kombinují, a provádějí tedy vyhlazení obrazu a detekci hran současně.

Přehled jednotlivých metod pro vyhlazování obrazu a detekci hran, jejich charakteristiky, výhody, nevýhody a příklady užití jsou uvedeny v kapitolách 8.4.5.

Podle vlastností funkčního vztahu pro výpočet výsledného jasu bodu na základě okolí ve vstupním obraze, dále metody předzpracování dělíme na lineární a nelineární. Lineární metody

počítají výsledný jas v bodě $g(i,j)$ jako lineární kombinaci jasů v lokálním okolí bodu $f(i,j)$ vstupního obrazu. Váha bodů v lokálním okolí f je určena koeficientem h :

$$f(i, j) = \sum_{(m,n) \in O} h(i - m, j - n) g(m, n) \quad (4.8)$$

Na tento vztah lze pohlížet také jako na diskrétní konvoluci s konvolučním jádrem h . V mnoha praktických případech se využívá pravoúhlého okolí O . Necht' je okolí O například obdélníkové o M řádcích a N sloupcích. Aby bylo použité okolí symetrické vůči středovému bodu $f(i,j)$, volí se nejčastěji rozměry M, N z množiny přirozených lichých čísel.

Lokální operace předzpracování obvykle jen málo využívají předběžné znalosti o povaze obrazu. Vzhledem k malému okolí zpracovávaného bodu je velice obtížné si tuto informaci odvodit i při vlastním průběhu zpracování. Zvláště pro vyhlazovací operace je vhodné znát charakter poruch v obraze, což mohou být například statistické parametry šumu, a podle nich přizpůsobit volbu metody a parametry zpracování.

Volba metody lokální transformace, velikosti a tvaru okolí zpracovávaného bodu, závisí také na velikosti objektů v obraze. Jsou-li objekty dostatečně velké, obraz může být vylepšen vyhlazením malých nespojitostí a velké objekty současně zůstanou zachovány.

Konvoluční lineární operace (filtry) jsou užitečné pro vyhlazování, detekci hran nebo čar. Lze poměrně jednoduše vytvořit specializované technické prostředky pro zrychlení výpočtu těchto operací. Zrychlení je jasně patrné pro speciální tvary filtrů. Příkladem jsou rekurzivní a separabilní filtry.

5 Segmentace obrazu

Jedním z nejtěžších kroků zpracování obrazu je jeho segmentace. Jedná se o analýzu obrazu vedoucí k nalezení objektů v obraze. Za objekty se zde považují části obrazu, které jsou bodem zájmu v průběhu dalšího zpracování. Cílem segmentace je tedy rozdělení obrazu do částí odpovídajících předmětům či oblastem z reálného světa. Výsledkem segmentace by měl být soubor oblastí, které odpovídají objektům ve vstupním obraze. Jedná se pak o tzv. kompletní segmentaci. Pokud ale oblasti neodpovídají přesně objektům, tak tuto segmentaci nazýváme částečnou. Kompletní segmentace obecně využívá vyšší úroveň zpracování, která je založena na znalostech řešeného problému. Částečná segmentace je založena na principu homogenity obrazových vlastností (např. jas nebo barva) uvnitř segmentu. Přesto existuje třída úloh, jejichž řešení umožňuje s dobrými výsledky i nižší úroveň zpracování. Častá je situace, kdy je obraz tvořen kontrastními objekty na pozadí neměnného jasu (krevní buňky, psaný text,...). Tam lze využít jednoduché globální postupy a dosáhnout kompletní segmentace obrazu na objekty a pozadí. Takové postupy nezávisí na obsahu, neuvádí se žádný model zpracovávané oblasti, k řízení procesu segmentace nepřispívají znalosti výsledné oblasti.

Při částečné segmentaci je výsledkem rozdělení obrazu do oddělených částí, které jsou homogenní vzhledem k zvolené vlastnosti jako je jas, barva, odrazivost, textura, atd. Po zpracování složitěho obrazu, jako je například letecká fotografie, je výsledkem seznam oblastí, které jsou homogenní v jistých rysech. Tyto oblasti se mohou překrývat. Částečně segmentovaný obraz musí být podroben dalšímu zpracování, které pomocí vyšší úrovně umožní získat výslednou segmentaci obrazu.

Úplně správná a kompletní segmentace složitých obrazů není v této fázi zpracování dosažitelná. Rozumným cílem je získat částečnou segmentaci, která je použita pro další zpracování operacemi vyšších úrovní. Okamžitým přínosem segmentace je výrazná redukce objemu zpracovávaných dat.

Jedním z hlavních problémů ovlivňujících segmentaci je nejednoznačnost obrazových dat, často doprovázená též informačním šumem. Segmentační metody lze rozdělit do tří skupin, a to podle dominantní vlastnosti, které je pro segmentaci využíváno. V první z nich je využíváno **globálních znalostí** obrazu reprezentovaných obvykle histogramem určitých vlastností. Druhá využívá **detekci hran**, třetí **detekci oblastí**. Druhá a třetí skupina tohoto třídění řeší duální problém, neboť každá oblast je jednoznačně reprezentována svou hranicí a každá uzavřená hranice jednoznačně vypovídá o oblasti, kterou určuje. V důsledku odlišného charakteru algoritmů tvorby hranic a oblastí přináší obě skupiny metod poněkud rozdílné segmentační výsledky, jinou informaci. Výsledky segmentačních postupů obou skupin lze proto kombinovat a vytvořit jedinou popisnou strukturu. Příkladem může být relační struktura, kde uzlům odpovídají oblasti, hranám relace sousednosti pomocí detekovaných hranic.

5.1 Prahování obrazu

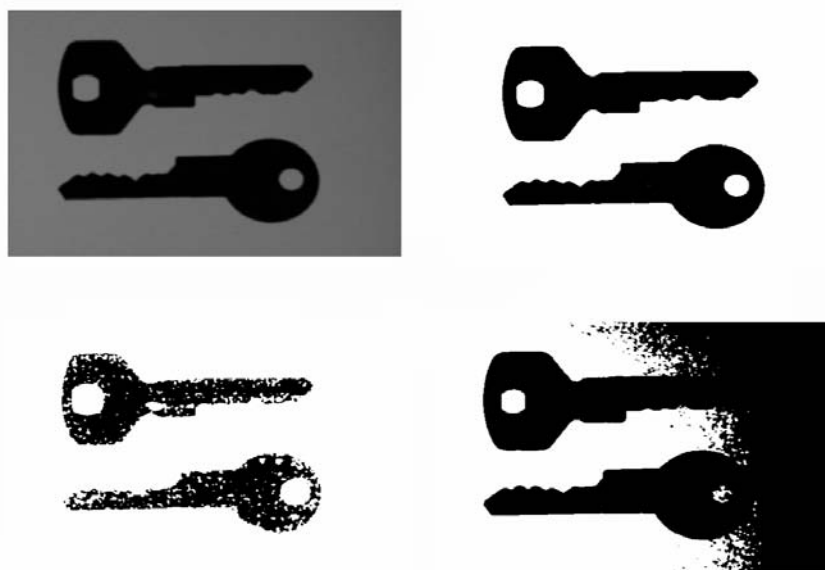
Prahování je nejjednodušším segmentačním postupem. Vychází ze skutečnosti, že mnoho objektů či oblastí obrazu je charakterizováno konstantní odrazivostí nebo pohltivostí svého povrchu. Pak je možno využít určené jasové konstanty – prahu, k oddělení objektů od pozadí. Prahování je nejstarší segmentační metodou a v jednoduchých případech je i nadále široce užíváno. Vzhledem k výpočetní nenáročnosti a jednoduchosti jej lze provádět i v reálném čase.

Prahování spočívá v transformaci obrazu na obraz binární, kde hodnotám jasu menším než hodnota prahu přiřadíme nulu, a hodnotám jasu větším než práh přiřadíme jedničku.

$$\begin{aligned} g(i, j) &= 1 & \text{pro } f(i, j) \geq T \\ &= 0 & \text{pro } f(i, j) < T \end{aligned} \quad (5.1)$$

Základní varianta prahování má mnoho modifikací. Jedna možnost je určit dvě hodnoty prahu, a poté rozdělit obraz na objekty – hodnoty jasu mezi dvěma prahy – a pozadí – hodnoty jasu příliš nízké nebo příliš vysoké. Takto provedené prahování se dá využít například segmentaci krevních buněk, kde interval mezi prahy reprezentuje hledanou cytoplasmu, pozadí je světlejší a buněčné jádro tmavší. Další zajímavá možnost jak využít tohoto způsobu prahování je jako velice jednoduchý detektor hran. Pokud máme tmavé objekty na světlém pozadí, tak některé hodnoty jasu, menší než pozadí a větší než objekt, se mohou vyskytovat pouze na hranici těchto objektů. Pokud tedy nastavíme interval právě na tyto hodnoty, zůstanou nám po prahování právě jen tyto hrany.

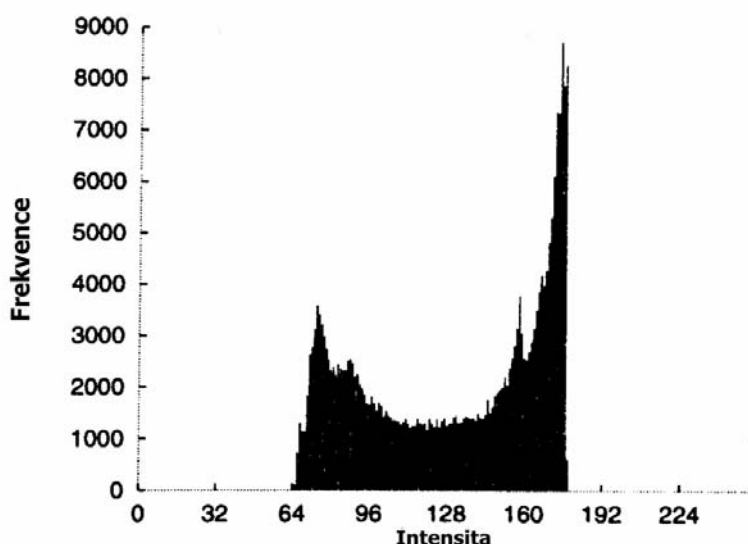
Další variantou je částečné prahování, kdy zvolíme hodnotu prahu, a poté hodnoty menší než tato hodnota označíme jako pozadí – přiřadíme hodnotu 0, a hodnotám větším ponecháme původní jas. Tím odstraníme z obrazu pozadí a vyskytující se objekty ponecháme nezměněné.



obr.5.1: prahování obrazu: původní obraz; segmentace prahováním; práh zvolen příliš malý; práh zvolen příliš velký

Při volbě hodnoty prahu nastává nejjednodušší situace v případě, je-li předem známa vlastnost, kterou má obraz po segmentaci mít. Práh určujeme tak, abychom tuto vlastnost segmentací splnili. Úlohou daného typu je segmentace stránky tištěného textu na znaky a pozadí, pokud víme, že písmena pokrývají $1/p$ plochy. S využitím znalosti poměru ploch objektů a pozadí je snadné určit (na základě histogramu) takovou hodnotu prahu T , aby právě $1/p$ plochy mělo úroveň jasu menší než T a zbytek mělo hodnotu jasu větší než T . Tato metoda se nazývá procentní prahování. Bohužel většinou informaci o procentním poměru objektů a pozadí nemáme. Tato informace může být někdy nahrazena jinou vlastností, známe například průměrnou šířku čar znaků. Práh můžeme zvolit tak, aby šířka čar v segmentovaném obraze odpovídala naší znalosti.

Složitější skupina metod volby prahu je založena na analýze tvaru histogramu. Obsahuje-li obraz objekty přibližně téhož jasu, které se odlišují od jasu pozadí, je výsledný histogram jasu dvouvrcholový (bi-modální). Jeden vrchol odpovídá jasu objektů, druhý vrchol potom odpovídá jasu pozadí. Takový histogram ukazuje obrázek 5.2.



obr.5.2: dvouvrcholový histogram

Z tvaru histogramu vyplývá, že hodnoty jasu ležící mezi oběma vrcholy nejsou v obraze příliš časté, a pravděpodobně odpovídají jasu hraničních bodů mezi objekty a pozadím. Výsledný práh by měl splňovat požadavek minimální segmentační chyby. Je proto logické určit práh pro oddělení objektů od pozadí jako hodnotu jasu, jejíž četnost je minimem ležícím mezi dvěma výše uvedenými maximy. V případě, že histogram je vícevrcholový (multi-modální), může být určeno více hodnot prahu, a to vždy v minimech mezi dvěma maximy. Pro každý práh je pak výsledek segmentace samozřejmě jiný.

Pro mnoho praktických úloh kde je využívána segmentace, je zapotřebí více informací, než je obsaženo v jediném spektrálním kanále. Pro získání většího množství informací je tedy daná scéna

snímána ve více barevných spektrech. Typickým příkladem jsou barevné obrazy, kde je informace kódována ve třech kanálech, červeném, zeleném a modrém. Dalším příkladem může být meteorologická satelitní fotografie, kde obraz může být snímán ještě ve více spektrech. Pro segmentaci takových obrazů je potom nutné provést multi-spektrální prahování. Prahování je provedeno nezávisle v každém spektrálním kanálu, a poté jsou jednotlivé výsledky složeny do jediného obrazu.

5.2 Detekce hran

Velkou skupinu segmentačních metod tvoří metody založené na informaci o hranách obrazu. Toto byl jeden z prvních způsobů segmentace obrazu a stále zůstává jedním z nejdůležitějších. Segmentace na základě detekce hran vychází ze skutečnosti, že hranice oblastí obrazu se sestávají z hran, které jsou v obraze nalezeny aplikací některého z hranových operátorů. Tyto nalezené hrany označují místa obrazu, v nichž dochází k nespojitosti v hodnotě jasu, barvy, nebo v textuře. Obraz, který vznikl aplikací hranového operátoru, je ale jako segmentační výstup většinou nepoužitelný. Proto musí po detekci hran následovat další zpracování obrazu hran, které spojuje hrany do řetězců lépe odpovídajících průběhu hranic objektů. Cílem je dosáhnout alespoň částečné segmentace obrazu, tzn. vytvořit z velkého množství hran obraz, v němž se budou vyskytovat jen takové hrany, jimž v původním obraze odpovídají hranice objektů nebo částí obrazu.

Jednotlivé segmentační metody se liší jednak strategií konstrukce výsledné hranice z hran, jednak množstvím předběžných informací, znalostí o zkoumaném obraze které je možné v konkrétní segmentační úloze využít. Čím více informací je při segmentaci k dispozici, tím lepší segmentační výsledky lze očekávat. Pomocí těchto informací lze také vyhodnotit věrohodnost výsledné získané segmentace. Tyto předběžné znalosti mají také vliv na vlastní průběh segmentace. Pokud je k dispozici velké množství znalostí o požadovaném výsledku, jsou jak celkový tvar hranice, tak i další vztahy k ostatním objektům v obraze určeny velice přísně a segmentační metoda musí všechny tyto dané podmínky dodržet.

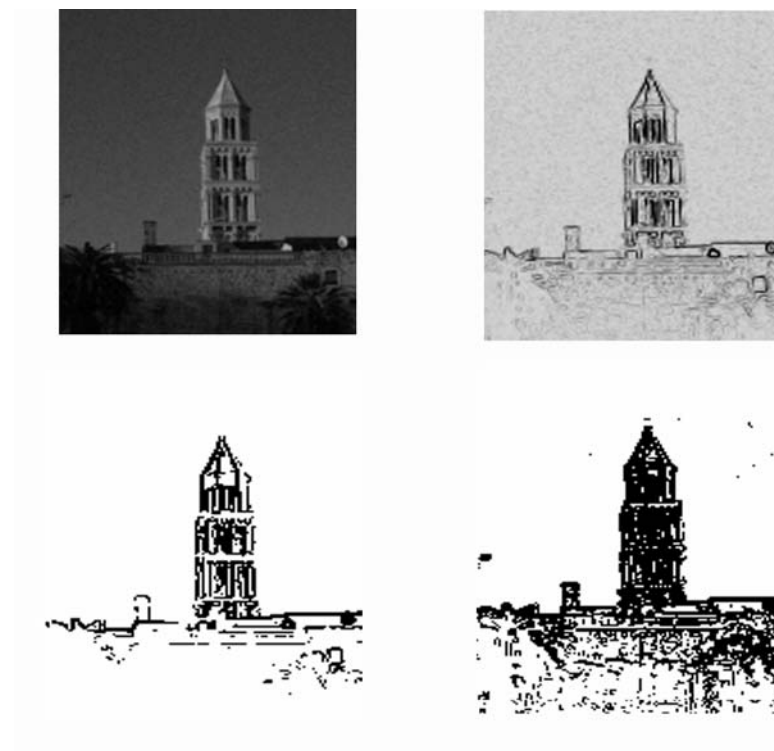
Naopak, pokud je málo informací o výsledné hranici, musí segmentační metoda brát v úvahu lokální vlastnosti obrazu spolu s obecnými znalostmi specifickými pro určitou aplikační oblast. Při malém množství předběžných znalostí také segmentační metody nemohou využívat hodnocení správnosti generované hranice a na jeho výsledku provádět zpětnovazební korekci segmentačního výsledku.

Nejčastějším problémem, se kterým se musí hranové segmentační metody vypořádat, je výskyt hran v místech bez přítomnosti skutečné hranice a současně absence hran tam, kde hranice ve skutečnosti probíhá. Tyto nepřesnosti jsou způsobeny šumem nebo neužitečnou informací v obraze. Je jasné, že obě tyto nepřesnosti nepříznivě ovlivňují výsledky segmentace.

Ve vytvořeném obraze hran je většinou jen několik málo míst s nulovou hodnotou velikosti hran. Malé hodnoty hrany odpovídají drobným změnám jasu vlivem kvantizačního šumu a malých nerovností v osvětlení. K odstranění těchto nevýznamných hran lze použít jednoduchou metodu prahování obrazu. Metoda vychází z obrazu hran, který je prahován vhodným prahem. Je-li zvolena větší hodnota prahu, pak se na výsledném obraze podílejí pouze významnější hrany. Hodnotu prahu lze určit například metodou procentního prahování. Jsou-li vstupní data dostatečně kontrastní a bez šumu, přináší tato metoda dobré výsledky.

K odstranění nevýznamných hran lze použít například také metodu hystereze. Tato metoda spočívá v určení dvou hodnot prahu, vysokého a nízkého. V obraze hran jsou potom všechny body, jejichž velikost je větší než vysoký práh, označeny za pravé hrany. Poté jsou zkoumány všechny body, jejichž hodnota leží mezi těmito dvěma prahy. Pokud je takový bod spojen s nějakou už nalezenou pravou hranou, je také označen jako pravá hrana.

Výhodné může být např. také vypuštění všech hranic, jejichž délka je menší než jistá hodnota.



obr.5.3: detekce hran: původní obraz; obraz po vyhledání hran; prahování hodnotou 30; prahování hodnotou 10



obr.5.4:detekce hran:obraz 5.3(b) po provedení hystereze z hodnotou vysokého prahu 70 a nízkého prahu 10

5.3 Srovnávání se vzorem

Srovnávání se vzorem je další základní metoda segmentace, která se používá k nalezení známých objektů v obraze nebo k hledání specifického vzoru. Srovnávání se může použít například pokud hledáme jeden objekt ve stejné scéně snímané z různých směrů. Jeden obraz může být použit k definování vzorů, a poté můžeme tyto (nebo podobné) vzory vyhledat v dalších obrazech.

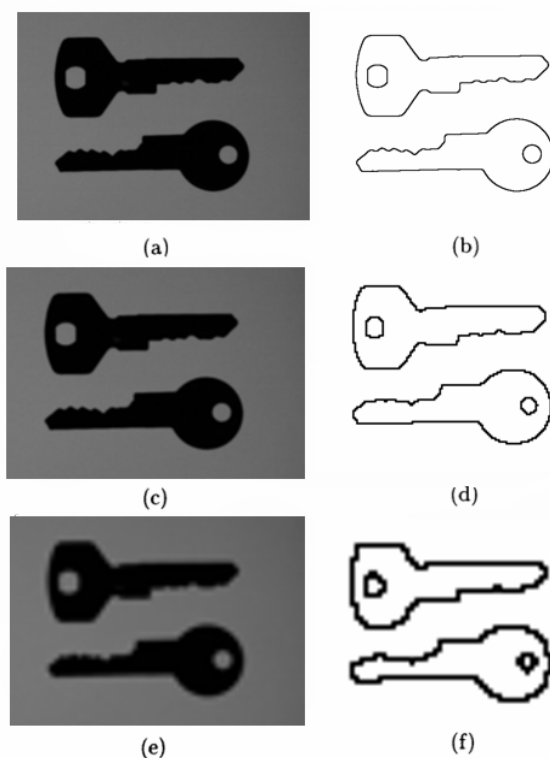
Srovnávací vzor může být velmi malý, nebo to může být celý objekt, který hledáme. Srovnávání se vzorem je velice snadné, pokud ve zpracovávaném obraze můžeme očekávat přesnou kopii původního vzoru. Většinou je ale část obrazu poškozená šumem, geometrickou deformací, atd. Proto nehledáme absolutní shodu, ale jen oblast kde je shoda se vzorem největší.

6 Popis objektů

Dalším krokem je popis obrazu nebo též popis nalezených objektů z předešlé segmentace. V této fázi se zkoumá tvar těchto objektů. Existují dva základní způsoby popisu. Jeden je založen na kvantitativním přístupu, což znamená popis objektů pomocí souboru číselných charakteristik. Mohou jimi být např. velikost objektu, jeho kompaktnost apod.. Druhou možností je kvalitativní přístup, ve kterém jsou popisovány relace mezi objekty a jejich tvarové vlastnosti. Způsob popisu je zvolen vždy podle toho, k čemu bude dál využit. Ve většině případů je tento popis vstupní informací pro rozpoznávání (klasifikaci) objektů. Výběr popisu je pak závislý na použitém rozpoznávacím algoritmu.

Definovat tvar objektu může být velice obtížné. Tvar je většinou vyjádřen slovně, a lidé používají výrazy jako podlouhlý, kulatý, s ostrými okraji apod. Počítačové zpracování obrazu však potřebuje mít tvar i velice složitých objektů popsán velice přesně. Dosud však není jasné, co je vlastně při popisu tvaru důležité a co ne, a existuje tedy spousta různých metod popisu tvaru objektu.

Dalším problémem při popisu tvaru objektu je velikost (rozlišení) obrázku, protože změna rozlišení může vyvolat změnu tvaru. Při vysokém rozlišení může dojít na hranici objektu vlivem šumu k změně hran, při nízkém rozlišení zase hrozí ztráta drobných detailů objektu. Vhodné rozlišení obrázku tedy musí být zvoleno s ohledem na charakter zkoumané scény.



obr. 6.1: a) původní obraz 640x480; b) hranice a; c) původní obraz 160x120; d) hranice c; e) původní obraz 64x48; f) hranice e

Je také nutné si uvědomit, že i když se zde jedná o 2D tvar a jeho popis, náš svět je tří-rozměrný, a ten samý objekt může z různých úhlů vypadat velice rozdílně. Ideální by proto bylo mít schopnost popsat tvar jakéhokoliv 3D objektu, žádný existující způsob ale takový popis složitých objektů neumožňuje. Je proto důležité zvážit, jestli si při popisu vystačíme s 2D popisem, nebo jestli je potřeba vzít do úvahy i třetí rozměr. Pro spoustu 3D objektů je 2D popis dostatečný, například je možné úspěšně identifikovat typ letadla podle jeho tvaru, i když se na něj díváme pod různými úhly. V mnoha dalších případech je ale nutné snímat objekt z určitého směru, který dává co nejvíce potřebných údajů pro identifikaci, například rozpoznání lidí podle snímku obličeje.

6.1 Identifikace oblastí

Identifikace jednotlivých oblastí je první krok nutný k popisu oblastí. Jedna z mnoha metod identifikace oblastí je označit každou oblast jednoznačným (celým) číslem. Nejvyšší číslo oblasti poté obvykle určuje i celkový počet oblastí v obraze. Tato metoda se nazývá barvení oblastí. Jiná metoda je užít menší množství identifikačních čísel (dostačující jsou čtyři), a ujistit se, že žádné dvě sousední oblasti nemají stejné identifikační číslo. Poté se ale při odkazování na tuto oblast musí uvést i nějaký bod, který do této oblasti náleží.

Předpokládejme, že segmentovaný obrázek R obsahuje m nezávislých oblastí R_i . Obrázek R většinou rozdělujeme na objekty a pozadí, kde R^C je doplněk, R_b je pozadí, ostatní oblasti jsou jednotlivé objekty.

$$R_b^C = \bigcup_{i=1, i \neq b}^m R_i \quad (6.1)$$

Vstupními daty pro barvicí algoritmus je většinou binární, nebo několikabarevný obrázek, kde pozadí je reprezentováno hodnotou nula a ostatní objekty nenulovými hodnotami. Jako výstup barvicího algoritmu je většinou zvolen několikabarevný obrázek, kde pozadí je reprezentováno nulovou hodnotou a jednotlivé objekty různými hodnotami odpovídající různým barvám.

Velice běžný problém který se vyskytuje při barvení oblastí a se kterým se musíme vypořádat, je kolize. Dochází totiž k tomu, že při procházení obrazu a hledání oblastí se nejdříve dvě oblasti jeví jako různé, a teprve později zjistíme, že jsou to dvě části jedné větší oblasti. Jednoduchá metoda, jak toto vyřešit, je provést barvení obrazu dvou-krokově. Ve druhém kroku potom nahradíme všechny tyto kolizní oblasti jednotnou barvou.

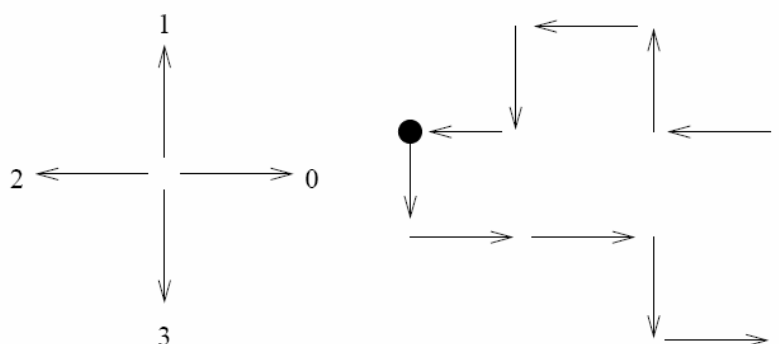
6.2 Popis objektů založený na popisu hranic

První metodou popisu objektů je metoda založená na popisu hranic objektů. Tuto hranici objektů musíme nějakým způsobem vyjádřit. Jednou možností jsou řetězové kódy. Řetězové kódy popisují objekt sekvencí jednotkových segmentů čar s danou orientací. Počáteční bod takové sekvence musí

být zadán samostatně, musí být určena jeho poloha v obraze. Výsledkem tohoto popisu je seznam čísel, která reprezentují hranu objektu. Řetězový kód může být snadno získán již při algoritmu detekce hran.

Pokud má být řetězový kód použit k srovnávání s nějakým vzorem, musí být nezávislý na volbě prvního bodu sekvence. Jedna možnost je najít jako první bod takový bod, aby výsledná sekvence, pokud ji interpretujeme jako jediné číslo, byla co nejmenší.

Řetězový kód je ale velice citlivý na vliv šumu a také změny měřítka a rotace objektu mohou způsobovat problémy.



obr.6.2: řetězový kód v 4-sousednosti

kód: 3, 0, 0, 3, 0, 1, 1, 2, 1, 2, 3, 2

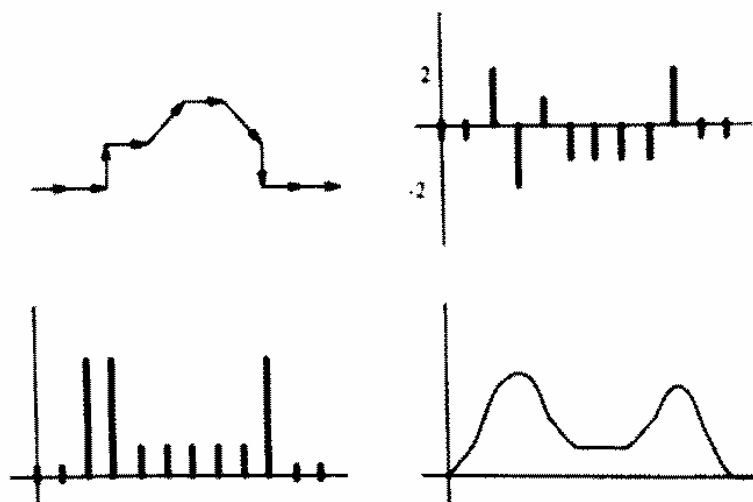
Vlastnosti kterými může být hranice objektu popsána, jsou délka hranice, křivost, ohybová energie, signatura.

Délka hranice je základní vlastnost objektu, která je jednoduše odvozena z jeho reprezentace řetězovým kódem. Délka každé vertikální a horizontální hrany je jedna, délka diagonální hrany (při uvažování 8-sousedství bodu) je $\sqrt{2}$. Délka hranice roste s rostoucím rozlišením obrazu.

Křivost objektu je v diskretním prostoru definována jako poměr mezi celkovou délkou hrany, a počtem bodů ve kterých hrana mění svůj směr. Čím menší je počet těchto bodů, tím přímější hranice je. Při výpočtu křivosti také většinou zjišťujeme, jak jsou které změny směru hrany zastoupeny, poté je můžeme například reprezentovat histogramem.

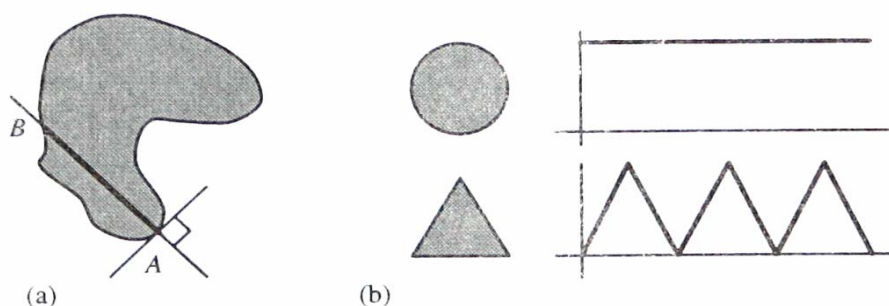
Ohybová energie hranice (křivky) může být chápána jako energie potřebná ke zkřivení přímky do požadovaného tvaru. Tato hodnota může být spočítána jako součet čtverců křivosti $c(k)$ na délku hrany L .

$$BE = \frac{1}{L} \sum_{k=1} c^2(k) \quad (6.2)$$



obr.6.3:ohybová energie: řetězový kód 0, 0, 2, 0, 1, 0, 7, 6, 0, 0; křivost 0, 2, -2, 1, -1, -1, -1, 2, 0; součet čtverců křivosti; vyhlazená křivka

Signatura tvaru objektu může být získána jako sekvence normálových vzdáleností objektu. Normálová vzdálenost je vypočítána pro každý bod jako vzdálenost k protějšímu bodu hranice. Signatury jsou ovlivněny šumem, používá se tedy k jejich určení vyhlazená hrana objektu.



obr.6.4:Signatura:konstrukce, signatury pro kruh a trojúhelník

6.3 Popis objektů založený na popisu plochy

Jednoduché objekty lze popsat podle vlastností, které má plocha tohoto objektu. Tento popis je jednoduchý, avšak lze jej použít jen na nepříliš složité objekty. Složitější objekty ale můžeme rozdělit na větší množství jednoduchých, a tyto popsat samostatně. Výhody tohoto způsobu popisu objektů jsou, že umístění a rotace objektu nemá žádný vliv, nejsou citlivé na malé změny tvaru, nejsou citlivé na šum, a jsou snadno srozumitelné.

Vlastnosti, které můžeme u objektu určit jsou: plocha, délka hranice, počet děr, plochy děr, kompaktnost, minimální a maximální vzdálenost hranice od těžiště, průměrná vzdálenost hranice od těžiště, poměr ploch děr k celkové ploše a momenty.

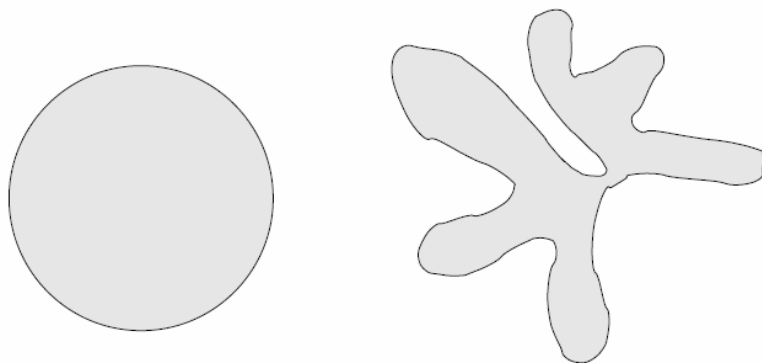
Plocha objektu je nejjednodušší a nezákladnější vlastnost oblasti. Určuje se podle počtu bodů, které do dané oblasti patří. Skutečná plocha bodu musí být známa, pokud chceme určit skutečnou plochu oblasti. Výpočet plochy se provede jednoduše, sečtou se všechny body, které do dané oblasti patří.

Délka hranice je počet hraničních bodů dané oblasti. Pokud se zvyšuje rozlišení obrazu, délka hranice konverguje k nekonečnu.

Kompaktnost je další oblíbenou vlastností používanou při popisu ploch, která je nezávislá na lineárních transformacích.

$$\text{kompaktnost} = \frac{(\text{délka hranice})^2}{\text{plocha}} \quad (6.3)$$

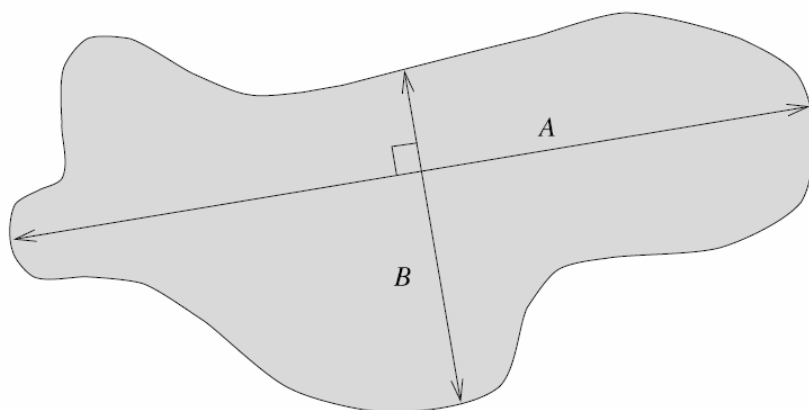
Nejkompaktnějším útvarem je kruh. Kompaktnost nabývá hodnot $[1, \infty)$.



obr.6.5: *kompaktní*

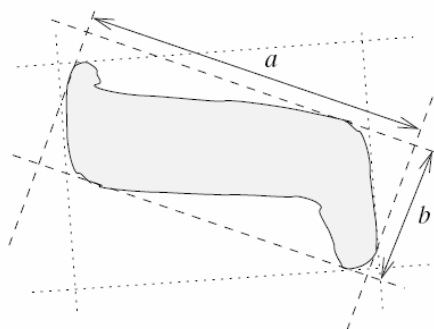
nekompaktní

Výstřednost je definována jako poměr mezi nejmenší a největší vzdáleností hranice od těžiště.

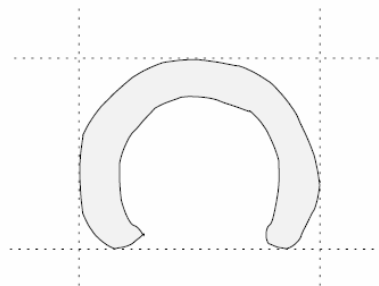


obr.6.6: výstřednost

Protáhlost je definována jako poměr mezi délkou a šířkou nejmenšího obdélníku, který ohraničuje námi zkoumaný tvar. Toto kritérium je použitelné jen u objektů, které nejsou příliš zakřivené. Jiné definice protáhlosti jsou založené na nejdelší přímce objektu, nebo ploše a počtu kroků *eroze*, které jsou nutné k tomu, aby obraz zmizel.



obr.6.6: protáhlost:použitelné



nepoužitelné

Eulerovo číslo je vlastnost, která je založena na počtu částí objektu S a počtu děr v objektu N . Objekt se může skládat z více nenavazujících částí, pokud se skládá z jedné části, je S vždy rovno jedné.

$$\mathcal{G} = S - N \quad (6.4)$$

Projekce horizontální a vertikální je definována jako:

$$p_h(i) = \sum_j f(i, j) \quad p_v(j) = \sum_i f(i, j) \quad (6.5)$$

Projekce může sloužit jako základ popisu obrazu založeném na ploše. Například šířka oblasti, ve které nejsou žádné díry, je definována jako maximální hodnota horizontální projekce oblasti.

Pravidelnost je definována jako poměr mezi plochou dané oblasti a plochou obdélníku ohraničující danou oblast.

Směr má význam jen u protáhlých objektů, kde je definován jako směr delší strany nejmenšího obdélníku, který ohraničuje danou oblast. Pokud jsou známy momenty objektu, směr lze spočítat podle vzorce:

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (6.6)$$

Poslední vlastnost, kterou můžeme určovat, jsou statistické momenty objektu. Momenty mohou být definovány nezávisle na velikosti, posunutí, rotaci. V digitalizovaných diskretních obrazech se velikost momentu spočítá podle vztahu:

$$m_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} i^p j^q f(i, j) \quad (6.7)$$

Centrální moment potom podle vztahu:

$$\mu_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} (i - x_c)^p (j - y_c)^q f(i, j) \quad (6.8)$$

7 Klasifikace objektů

Finálním krokem při zpracování obrazu je klasifikace (rozpoznání obrazu). Ve většině případů se jedná o zařazení objektů nalezených v obraze do skupiny předem známých tříd. Metody klasifikace objektů se dělí do dvou základních skupin, které jsou úzce spjaty se způsobem popisu objektů. Jedná se o příznakové rozpoznávání a strukturální rozpoznávání.

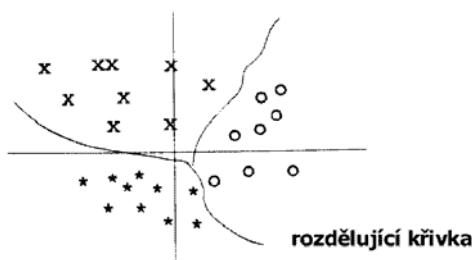
Příznakové metody jsou založeny na principu využití příznaků, což je skupina číselných charakteristik objektu. Učení vlastního klasifikátoru zde může být s trénováním množinou i bez ní, na principu shlukové analýzy.

Strukturální rozpoznávání využívá jako vstupu kvalitativní popis objektů. Objekty jsou zde popsány primitivami. Dále je definována abeceda, jazyk popisu a gramatiky jednotlivých tříd. Vlastní rozpoznávání je pak založeno na principu rozboru slova a kontroly správnosti syntaxe pro jednotlivé třídy.

7.1 Příznakové rozpoznávání

Příznakový popis předmětu se vyznačuje číselným charakterem elementárních popisů. Elementární popisy nazýváme příznaky. Tyto příznaky jsou například výsledkem práce některých metod popisu předmětu popsaných v předešlé kapitole. Množina všech možných obrazů vytváří n -rozměrný prostor, který se nazývá obrazovým prostorem. Při vhodném výběru příznaků je podobnost předmětů v každé třídě vyjádřena geometrickou blízkostí jejich obrazů v obrazovém prostoru. Jednotlivým třídám odpovídají shluky obrazů, které lze zhruba oddělit vhodnou křivkou (nebo u vícerozměrných obrazů vhodnou nadplochou). Existuje-li pro danou úlohu klasifikace taková rozdělovací křivka, že v každé z oblastí leží vždy jen obrazy jedné třídy, jde o úlohu se separabilními množinami obrazů. U separabilních úloh může každá část obrazu reprezentovat výhradně předměty jedné třídy. Lze očekávat, že u separabilních množin obrazů lze dosáhnout bezchybné klasifikace.

Velká část úloh ale nemá separabilní množiny. V takovém případě musí umístění rozdělovací křivky v obrazovém prostoru představovat určitý kompromis. Oblasti nelze vymezit tak, aby v každé z nich byly jen obrazy ze stejné třídy, proto je jistá část předmětů vždy chybně klasifikována.



obr. 7.1: rozdělovací křivka

Klasifikátor je funkce, které pro předmět popsany příznaky rozhodne, do které třídy předmět patří. Klasifikátory můžeme dále rozdělit na lineární a na klasifikátory pracující na principu minimální vzdálenosti.

Při určování správného nastavení klasifikátoru, je obvykle známa množina vybraných předmětů, u kterých je známé jejich správné zařazení do třídy. Takovou množinu nazýváme trénovací množinou. Tato trénovací množina je v praxi vždy konečná. Použitelné metody pro nastavení klasifikátoru musí být takové, že informaci získanou z prvků trénovací množiny zobecní na celý obrazový prostor takovým způsobem, že výsledné nastavení klasifikátoru bude optimální pro všechny možné obrazy a ne jen pro obrazy obsažené v trénovací množině. Klasifikátor tedy musí správně rozpoznávat i obrazy, se kterými se nikdy předtím nesetkal.

Čím větší je tato trénovací množina, tím přesnější výsledné nastavení klasifikátoru bude. V praxi je tedy tato trénovací množina dodatečně rozšiřována, až dokud nebude dosaženo požadované přesnosti nastavení klasifikátoru. Dochází tedy k procesu učení klasifikátoru na dané vstupní obrazy.

7.2 Syntaktické rozpoznávání

Zatímco v příznakových metodách rozpoznávání je užíván kvantitativní popis předmětů číselnými parametry, v syntaktických metodách má vstupní popis kvalitativní charakter odrážející strukturu předmětu. Syntaktický popis je využitelný v případě, když příznakový popis není schopen obsáhnout velkou složitost popisovaného předmětu, a když je vhodné předmět popsat hierarchickou strukturou složenou z jednodušších částí. Elementární vlastnosti se nazývají primitiva. Poté co každé primitivum označíme symbolem a popíšeme vzájemné vztahy mezi primitivy pomocí relací, získáme popis relační strukturou.

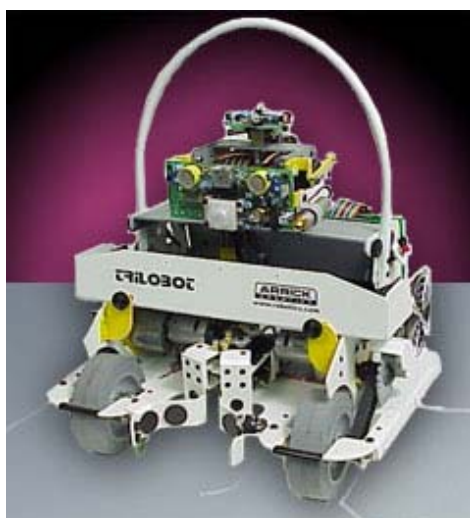
Při výběru primitiv je vhodné, aby byl jejich počet co nejmenší a současně aby umožňoval vytvořit odpovídající syntaktický popis obrazu. Primitiva by měla být z obrazu snadno extrahovatelná a klasifikovatelná a měla by odpovídat přirozeným a výrazným elementům struktury zpracovávaného obrazu.

Množina všech primitiv je potom nazývána abecedou. Množina všech slov nad danou abecedou, pomocí nichž lze popsat předměty jedné třídy, se nazývá jazyk. Jazyk v sobě zahrnuje popisy všech předmětů dané třídy. Gramatika poté představuje soubor pravidel, pomocí nichž lze z prvků abecedy vytvářet slova náležející do daného jazyka.

8 Návrh programu

8.1 Robot „Trilobot Research Robot“

Jako předmět rozpoznávání byl vybrán robot Trilobot firmy Arrick Robotics. Jedná se o malého, přenosného, mobilního robota, který je určen převážně k universitnímu výzkumu. Jeho rozměry jsou 30x30x30cm, váha asi 5kg.



obr.8.1: robot Trilobot, pohled zepředu

Je vybaven dvěma předními a jedním zadním říditelným kolem, jeho maximální rychlost je asi 25cm/s. Poháněn je diferenciálním elektrickým motorem a napájen osmi bateriemi.

Dále je vybaven držadlem a robotickou rukou umožňující uchopit malé plechovky a míče. Ovládán může být prostřednictvím infračerveného dálkového ovládání, rádiového ovládání, nebo prostřednictvím kabelu. Pro komunikaci s počítačem využívá standardní sériový port (RS-232).

Co se týče sensorů, tak je robot vybaven osmi dotykovými čidly po stranách, dále elektronickým kompasem k určování směru. Dále sonarem umožňujícím detekovat objekty a jejich vzdálenost, pasivním infračerveným pohybovým detektorem umožňujícím zjistit přítomnost lidí, čtyřmi sensory reagujícími na světlo, teploměrem, čidlem na nárazy, a čidlem na vodu.

Jeho využití je převážně pro výuku robotiky, experimentů s umělou inteligencí, výzkum navigace, a další vědecké projekty.

Elektronický kompas a sensory pro určování polohy, kterými je robot vybaven, pracují však velice nepřesně, což byl také jeden z důvodů pro vznik tohoto programu, který by určoval polohy a orientace robotů v laboratoři.

8.2 Příprava robotů a laboratoře

Pro návrh vlastního programu, který by rozpoznával umístění a orientaci robotů v laboratoři, bylo nejprve nutno připravit fotografie, na kterých by byli roboti rozpoznatelní.

V rámci předešlého ročníkového projektu jsem řešil podobný úkol, ale s tím rozdílem, že robot byl v obraze pouze jeden. S dalšími roboty vyvstala tedy otázka, jak je identifikovat, jak je od sebe vzájemně rozlišit.

Po důkladné prohlídce robotů, kdy jsem zjistil, že jsou všichni naprosto stejní a nemají na sobě žádné identifikační znaky, jsem se tedy rozhodl, že je nějak musím označit, abych je mohl jednoznačně identifikovat. Po prozkoumání a zvážení možností jaké znaky a kam na roboty umístit, jsem se rozhodl pro označení robotů kruhovými tečkami. Tečky jsem zvolil různé barevné, a to tak, abych mohl podle tří teček, které umístím na každého robota, jej jednoznačně identifikovat.

Uvažil jsem, že budou stačit čtyři různé barvy teček. Pro rozlišení robota bude zapotřebí nejen určitá kombinace těchto čtyř barev, ale i určité pořadí teček. Zvolil jsem tedy červenou, zelenou, modrou a žlutou barvu.

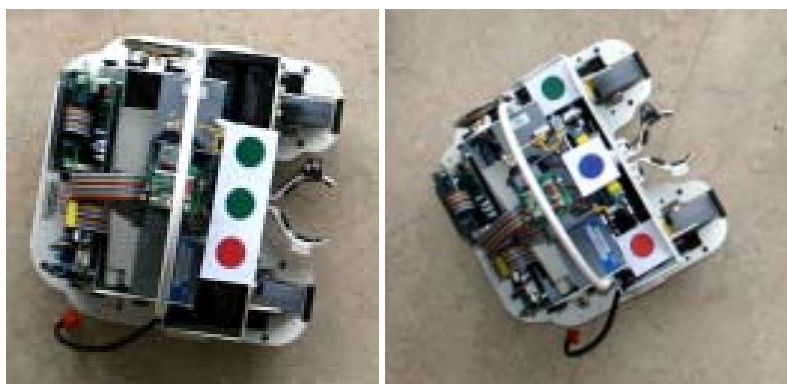
S přihlédnutím k velikosti robota (30 cm x 30 cm) jsem zvolil velikost teček. Rozhodl jsem se pro tečky o průměru 3 cm. Tato velikost je dostatečná pro to, aby byly tečky dobře vidět, ale zároveň není ještě příliš velká, takže jdou snadno na roboty umístit.

Jako vhodné místo pro umístění těchto teček, tak aby byly při pohledu shora dobře patrné, mě napadly hned dvě možnosti, rozhodl jsem se tedy vyzkoušet obě dvě.

Jako první možnost jsem tyto tři značky umístil na proužek papíru, který jsem umístil celý na „hlavu“ robota.































Jako druhou možnost jsem umístil tyto značky na levý a pravý okraj robota – na baterie, a třetí značku doprostřed – na „hlavu“ robota.

Polohy značek jsou dobře patrné na obrázku 8.2.



obr.8.2:umístění značek: (a) první možnost, (b) druhá možnost

Barevné kombinace, které určují jednotlivé roboty, jsem poté zvolil pro každé ze dvou možných umístění rozdílné. Vznikly tak dvě různé identifikační sady značek. První sada rozlišuje čtyři roboty, druhá sada rozlišuje šest robotů. Jednotlivé barevné kombinace ukazuje tabulka 8.3.

	Sada 1		Sada 2	
Pořadí	Znaky	Popis	Znaky	Popis
1	  	č-č-z	  	č-z-m
2	  	č-z-z	  	č-z-ž
3	  	č-m-m	  	č-m-z
4	  	č-ž-ž	  	č-m-ž
5			  	č-ž-z
6			  	č-ž-m

tabulka: 8.3: dvě identifikační sady

Při pohledu do této tabulky je hned patrné, že jako první značka byla ve všech případech použita červená tečka. Tuto možnost jsem zvolil z toho důvodu, aby bylo vždy jednoznačně jasné pořadí těchto tří teček. Toto jednoznačně dané pořadí poté umožní, aby byla podle těchto teček rozpoznatelná i orientace robota.

Dále je zřejmé, že pro první identifikační sadu jsem vždy zvolil dvě tečky stejné barvy, v druhé sadě má potom každá značka jinou barvu.

Nyní tedy máme už všechny roboty označené a připravené k rozpoznávání. Pro splnění zadání však bylo ještě nutné umístit do obrazu nějaký další předmět, který by roboti byli schopni uchopit do robotické ruky. Vzhledem k velikosti připadalo v úvahu použít buď plechovku, nebo nějaký malý míček. Po zvážení možností jsem jako další předměty tedy zvolil tři pěnové míčky různých barev, které se jinak používají při plážovém tenise. Tyto míčky se běžně prodávají v červené, zelené a žluté barvě, rozhodl jsem se nakonec použít všechny tři. Míčky jsou v obraze velice výrazné a jasně rozpoznatelné, ukazuje je obrázek 8.4.



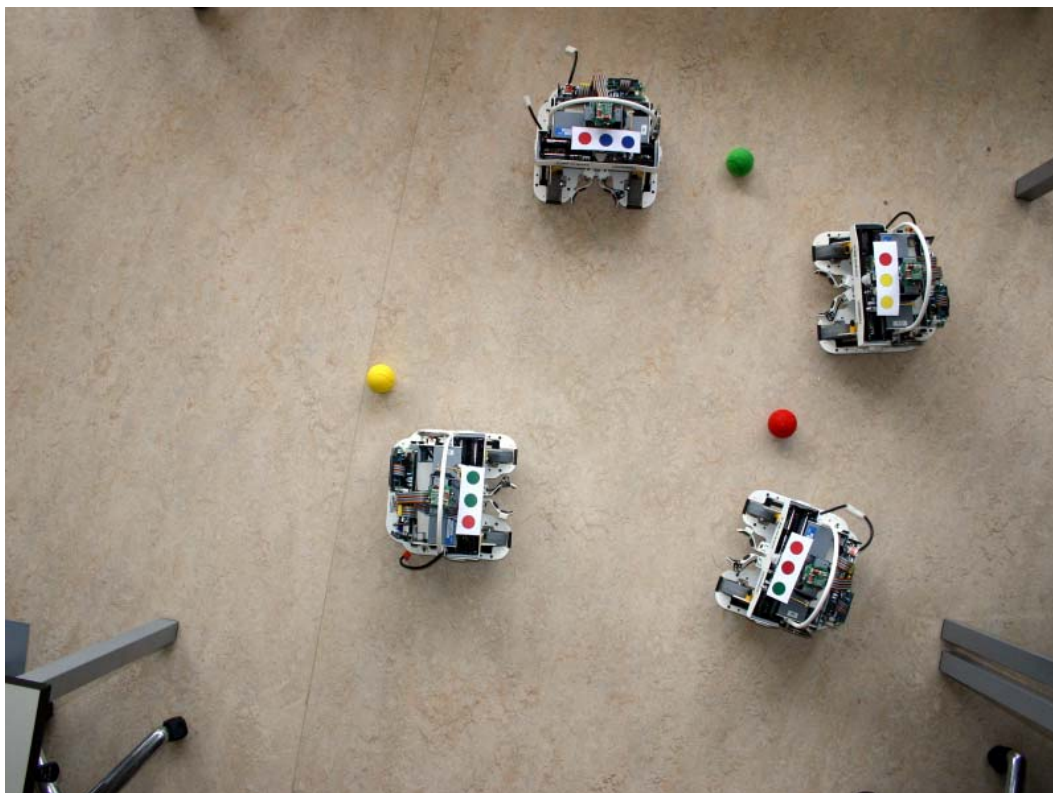
obr.8.4: míčky které se budou rozpoznávat

Po dokončení přípravy bylo tedy možné přistoupit k vlastnímu nafocení robotů. Roboti a míčky byli umístěny v laboratoři, kde se jinak žádné další zvláštní přípravy neprováděly. To znamená, že na výsledných snímcích jsou kromě objektů, které se mají rozpoznat, i další předměty, jako například části stolů či židlí. Podlaha v místnosti byla z linolea, poměrně nekонтastní běžové barvy, takže rozpoznávání robotů nebránila.

Snímky jsem fotil digitálním fotoaparátem, z ptačí perspektivy. V laboratoři není zatím instalováno žádné zařízení, které by umožňovalo fotoaparát připevnit ke stropu a pořizovat snímky automaticky, takže jsem využil přenosné schůdky a snímky fotil z ruky, s pomocí stativu. Na některých snímcích jsou tedy tyto schůdky částečně vidět.

Všechny fotografie byly pořízeny z výšky asi 2 metrů a přibližně ze stejného místa. Snímáním scény z ptačí perspektivy se minimalizuje 3D zkreslení objektů. Výška 2 metry je však ještě poměrně málo. Aby bylo toto zkreslení nerozpoznatelné, bylo by nutné celou zkoumanou scénu snímat z podstatně větší výšky.

V laboratoři jsem tedy nafotil sérii asi dvaceti fotografií, na kterých byly objekty vždy jinak umístěné, s rozdílným natočením. Na první části fotografií jsou čtyři roboti označení první identifikační sadou, na druhé části je potom šest robotů označených druhou sadou. Na některých snímcích je jeden z robotů částečně mimo obraz. Část fotografií je pořízena s pomocí blesku, jiné bez blesku, vyskytují se na nich tedy rozdílné světelné podmínky.



obr.8.5: *prohledávaná scéna: robot, míčky a další objekty*

8.3 Digitalizace obrazu

Jako nástroj pro vytvoření vlastního programu na rozpoznávání objektů v obraze jsem zvolil vývojové prostředí Borland Developer Studio 2006, z kterého jsem vybral jazyk Delphi for Win32. Toto vývojové prostředí poskytuje dostatečné vývojové a další podpůrné prostředky pro vytvoření, ladění a kompilaci aplikací pro Windows. Dále nabízí některé základní prostředky pro práci s bitmapovou grafikou (načítání bitmapových dat, jejich zobrazení, práci s jednotlivými body obrazu), čehož jsem dále také využil.

Nyní k vlastní digitalizaci. Funkce jazyka Delphi pro práci s bitmapovou grafikou jsou dostupné především prostřednictvím komponenty TImage. Tato komponenta umožňuje nahrávání grafických dat ze souboru, jejich zobrazení na monitoru, a především lze snadno přistupovat přímo k jednotlivým bodům obrazu přes jejich souřadnice.

Jeden z několika podporovaných grafických formátů který lze přímo načítat komponentou TImage je soubor BMP. Jako typ vstupního souboru se kterým bude program pracovat, jsem tedy zvolil tento formát, jako rozlišení vstupního souboru jsem zvolil rozměr 800 x 600 bodů. Toto rozlišení jsem zvolil především z toho důvodu, že jednak poskytuje dostatečné množství detailů v obraze, nedochází ke ztrátě objektů ani hran, a jednak je práce s takovým malým rozlišením poměrně rychlá a nenáročná na paměť.

Po výběru a otevření vstupního souboru dojde tedy k jeho načtení do paměti, a rozdělení na jeho tři barevné složky, Red, Green a Blue, se kterými můžeme dále samostatně pracovat. Hodnoty jasu jednotlivých barevných složek se uloží do tří nezávislých polí. Toto rozdělení a uložení do polí je provedeno z toho důvodu, že poté jednak můžeme, podle svého uvážení, pracovat buď s jednou barevnou složkou, nebo s jejich libovolnou kombinací, a jednak je práce se třemi samostatnými poli mnohem jednodušší a rychlejší než přístup k jednotlivým bodům přímo přes komponentu TImage.

```
c := Image1.Canvas.Pixels[x,y];  
rmatrix[x,y] := (c and $0000FF);  
gmatrix[x,y] := (c and $00FF00) shr 8;  
bmatrix[x,y] := (c and $FF0000) shr 16;
```

Z ukázky kódu je patrné, že jednotlivé barevné složky jsou získány ze složené hodnoty určující barvu bodu, kde vpravo je hodnota červené složky, uprostřed zelené složky, a vlevo modré barevné složky. Dále je ještě třeba podotknout, že souřadnice bodů x, y jsou značeny od levého horního okraje obrazu.

Nyní tedy máme k dispozici tři různé pole, každé o rozměrech 800 x 600 bodů, které v každém svém bodě obsahuje číslo v rozmezí 0-255, které udává hodnotu jasu dané barevné složky tohoto bodu. S těmito poli budeme dále pracovat.

8.4 Předzpracování obrazu

Nyní již máme k dispozici digitalizovaný obraz, můžeme ho tedy začít zpracovávat. V této fázi, předzpracování obrazu, jsem se rozhodl provést několik různých transformací. Všechno to jsou transformace, které nějakým způsobem mění obraz, nějaké jeho vlastnosti potlačují a nějaké zvýrazňují. Ve výsledku je obraz čitelnější, jasnější a jednodušší. Jako první jsem zvolil metodu jasových transformací obrazu, a to prahování. Před vlastním prahováním, pro účely správné hodnoty prahu, bylo ale nejprve třeba získat histogramy obrazu.

8.4.1 Tvorba histogramu

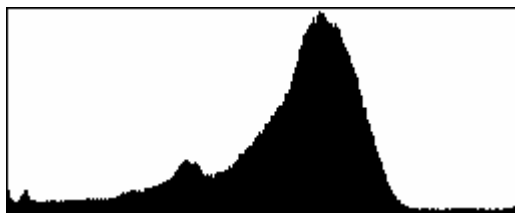
Histogram jasu obrazu ukazuje frekvenci výskytu jednotlivých hodnot jasu v obraze. Histogram obrazu který obsahuje L úrovní jasu je vyjádřen jednorozměrným polem o L prvcích.

Algoritmus pro výpočet histogramu:

1. Přiřaď hodnotu 0 do všech prvků pole h_f .
2. Pro všechny body (x, y) obrazu f , přičti k $h_f[f(x,y)]$ jedničku.

Histogram je většinou jedinou globální informací o obrazu, kterou můžeme získat. Je nutné si uvědomit, že jednomu histogramu může odpovídat několik obrazů. Například při změnách poloh objektu v jinak neměnném obraze zůstane histogram vždy stejný.

Nejčastější způsob prezentace histogramu je jako sloupcový graf. Takový histogram je ukázán na obrázku 8.6.



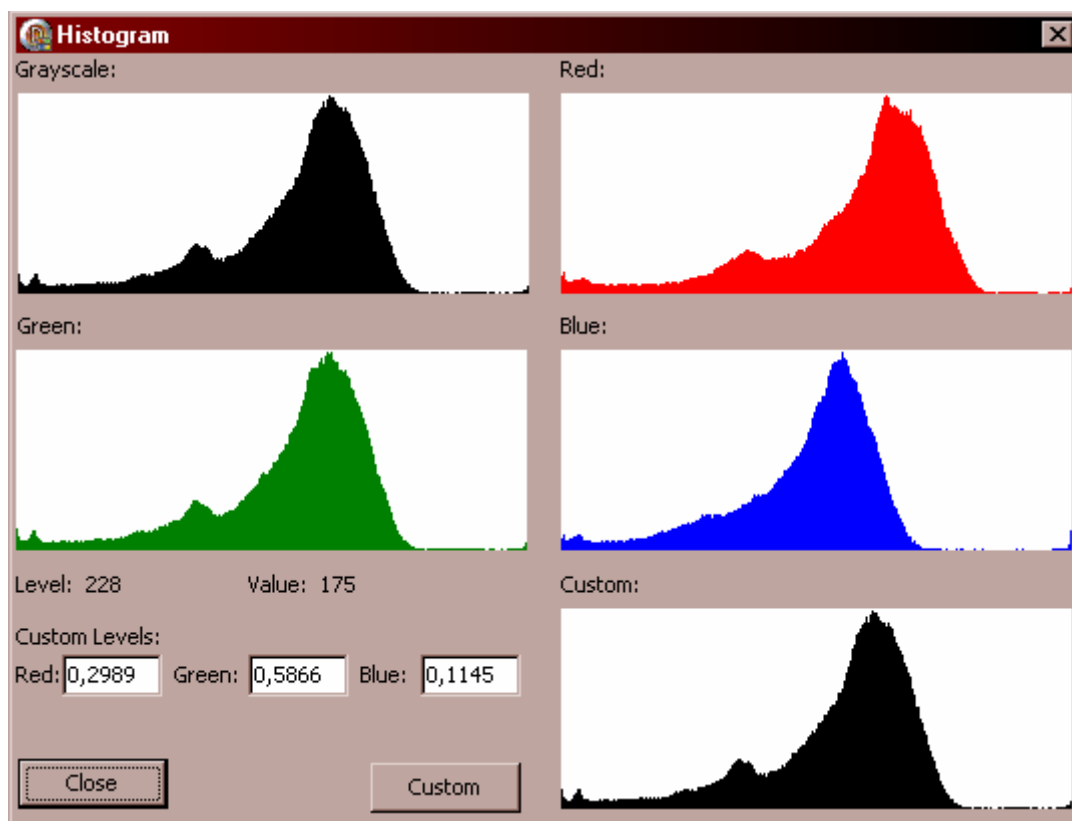
obr.8.6: *histogram obrazu*

Z tohoto histogramu můžeme vyčíst, že v obraze téměř nejsou zastoupeny jasnější tóny, také tmavší tóny jsou v obraze zastoupeny poměrně málo. Je zde vidět jedno výrazné maximum, zhruba ve středu rozsahu, které odpovídá nejčastější hodnotě jasu. Obraz tedy celkově není příliš kontrastní.

Jelikož nezpracováváme snímky černobílé ale barevné, máme několik možností jak vytvořit histogram. První možnost je vytvořit tři histogramy, pro každou barevnou složku zvlášť.

Druhou možností je vytvořit jeden histogram, který zpracuje všechny tři barevné složky společně. Musíme si ovšem uvědomit, že tyto tři složky nemůžeme jednoduše zprůměrovat, musíme

přihlédnout k vlastnostem lidského zraku, které každou složku vnímá s jinou intenzitou. Tento poměr je vyjádřen vztahem (3.2).



obr.8.7: dialog tvorby histogramu

Na obrázku 8.7 vidíme jednotlivé histogramy pro zpracovávaný obraz. Jsou zde zobrazeny tři histogramy pro každou barevnou složku zvlášť a složený histogram. Máme také možnost zadat vlastní poměr jednotlivých barevných složek v histogramu.

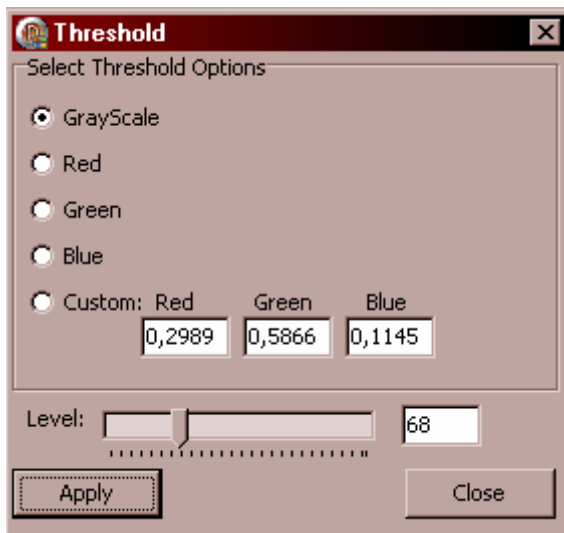
Vidíme zde, že histogramy jednotlivých složek jsou vzájemně odlišné, tyto odchylky však nejsou nijak výrazné, zvlášť u červené a zelené složky, které jsou pro lidské vnímání nejdůležitější.

Po zhodnocení těchto histogramů je tedy možno přejít k vlastnímu prahování.

8.4.2 Prahování obrazu

Prahování slouží k oddělení objektů od pozadí. Je prováděno podle vzorce (5.1). Pro hodnotu jasu ve vstupním obraze menší než je zvolená hodnota prahu se ve výstupnímu obraze přiřadí hodnota nula, pro hodnotu větší pak jednička.

Toto prahování probíhá pro každou barevnou složku zvlášť (případně pro složený obraz), a se zvolenou hodnotou prahu, která je vhodná pro právě vyhledávaný objekt v obraze. Je zde také možnost nastavit vlastní poměr váhy barevných složek.



obr.8.8: prahování

```

svetlost := trunc(rmatrix[x,y]*redP+gmatrix[x,y]*greenP+bmatrix[x,y]*blueP);
if svetlost > prah then tmatrix[x,y] := color
                    else tmatrix[x,y] := clBlack

```



obr. 8.9: prahovaný obraz

Na obrázku 8.9 vidíme výsledek prahování snímané scény. Jedná se o prahování složeného černobílého obrazu (jsou obsaženy všechny barevné složky), hodnota prahu je zvolena 65. Po zhodnocení můžeme konstatovat, že roboti jsou v obraze dobře patrní, avšak netvoří jeden kompaktní

celek, který by bylo snadné identifikovat. Určit hranice těchto robotů také není snadné, viz. např. robot uprostřed dole, který vrhá stín a nelze tak dobře rozpoznat, co je ještě robot a co už je jen jeho stín.

Identifikační znaky nejsou téměř vůbec patrné. Pro zobrazení těchto znaků můžeme zkusit experimentovat s různým nastavením prahovaných barevných složek, případně jejich kombinací, a s různou hodnotou prahu. Po provedení těchto experimentů však lze říci, že nepřinesly očekávané výsledky.

Identifikační tečky se sice podařilo při některých nastaveních hodnot zvýraznit, avšak toto zvýraznění nebylo dostatečné, současně došlo také ke zvýraznění mnoha ostatních objektů v obraze. Znaky pak nebylo možno v obraze nalézt. Také se přes poměrně široké rozmezí zkoumaných hodnot nepodařilo zvýraznit všechny tečky, některé, zvláště v tmavějších místech scény, zůstávaly stále skryté.

To je způsobeno tím, že prahování zkoumá hodnoty jasu. A i když vezmeme jen určitou barevnou složku, tak v tmavších místech obrazu bude mít tato složka poměrně malé hodnoty, mnohem menší než ve světlejších částech.

Další objekty, které jsou v obraze, jsou míče, které má robot uchopit. Tyto míče nejsou však v prahovaném obraze téměř vůbec vidět, situace je podobná jako v případě identifikačních značek. Je zde akorát ten rozdíl, že míče jsou tří-rozměrné objekty, které vrhají stín. Tento stín lze v prahovaném obraze vcelku dobře rozpoznat, samotné míče však nikoli.

Stejně jako v případě identifikačních teček experimentování s různými hodnotami nepřináší žádané výsledky, vysvětlení je stejné.

Dalšími objekty, které můžeme v obraze nalézt jsou části stolů a židlí. Tyto objekty jsou poměrně dobře rozpoznatelné, a to z toho důvodu, že různé hrany těchto objektů jsou odlišně natočeny ke zdroji světla, tudíž na nich vznikají oblasti stínů a osvětlených míst, které se na prahovaném obraze dobře projeví.

Jediným problémem u těchto objektů jsou potom stíny, které vznikají pod nimi, a které samozřejmě k tomuto objektu nepatří. Tento jev je dobře patrný u stolů a židlí v levé spodní a pravé spodní části obrazu.

Po zhodnocení těchto výsledků prahování můžeme tedy říci, že v tomto typu úlohy je jen málo použitelné. Výsledky můžeme využít k nalezení míst v obraze „kde něco je“, nějaká překážka, avšak problém by způsobily stíny, které se vyskytují i v místech, kde je prázdný prostor. Problém by byl také s míči, které jsou jen špatně rozpoznatelné.

Co se týče robotů, tak metoda prahování by šla použít pro nalezení jednotlivých robotů, ne však už pro jejich úspěšnou identifikaci.

Další metodou, kterou se budu zabývat, je tedy taková transformace obrazu, která by vedla k nalezení a zvýraznění jednotlivých barevných složek v obraze.

8.4.3 Zvýraznění barev v obraze

Další prováděnou úpravou, kterou jsem se rozhodl implementovat, je tedy vyhledání a zvýraznění určitých barev v obraze. K tomuto kroku jsem se rozhodl po neúspěchu prahovací metody odlišit v obraze jednotlivé identifikační znaky a barevné míče.

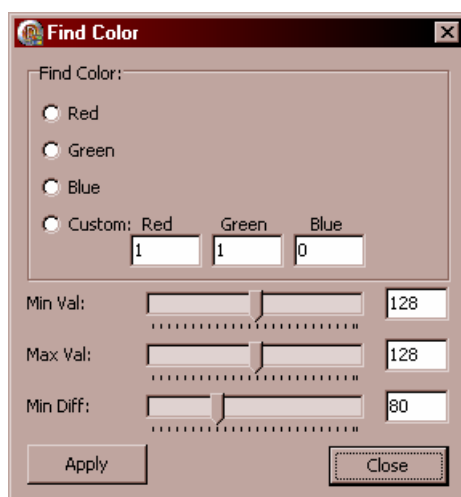
Barva, kterou vnímáme obraz, je dána vzájemným podílem jasů jednotlivých barevných složek v každém bodě. Úkolem je tedy určit, jaký je takový rozsah poměrů, který ještě vnímáme jako určitou barvu, kde je hranice mezi jednotlivými barvami.

Identifikační značky se v obraze vyskytují v červené, zelené, modré a žluté barvě. Míče jsou červený, zelený a žlutý. Zaměřím se tedy na zvýraznění právě těchto barev, ostatní barvy pro nás v tomto okamžiku nejsou zajímavé.

Aby mohlo být provedeno úspěšné zvýraznění barev, tak je jako první potřeba určit kombinace barevných složek které ukazují na jednotlivé barvy. Pro správné určení těchto kombinací jsem experimentoval s nastavováním poměrů, minimální hodnotou klíčové složky jakou musí být určitá barva, maximální hodnotou barevných složek barev, které by se ideálně v dané barvě vůbec neměly vyskytovat, a minimálním rozdílem mezi těmito hodnotami.

V naprosto ideálním případě by červená barva měla obsahovat jen červenou složku, zelená barva jen zelenou, modrá jen modrou a žlutá barva by měla obsahovat jen červenou a zelenou složku přibližně ve stejném poměru. Ostatní barevné složky by se v těchto barvách vůbec neměly vyskytovat.

Ve skutečném snímku ale barvy tak jasně rozpoznatelné nejsou, mají různé odstíny, hodnoty jednotlivých složek se mohou měnit v poměrně značném rozsahu.



obr.8.10: vyhledávání barvy podle podílu jednotlivých složek

Po provedení experimentů lze říci, že rozmezí jednotlivých složek je pro každou barvu jiné, například červená barva je velice snadno určitelná a nezaměnitelná s jinými barvami, ale třeba zelená

barva může plynule přecházet v zeleno-modrou a poté až modrou, a je tedy obtížné určit dělicí hranici mezi těmito dvěma barvami.

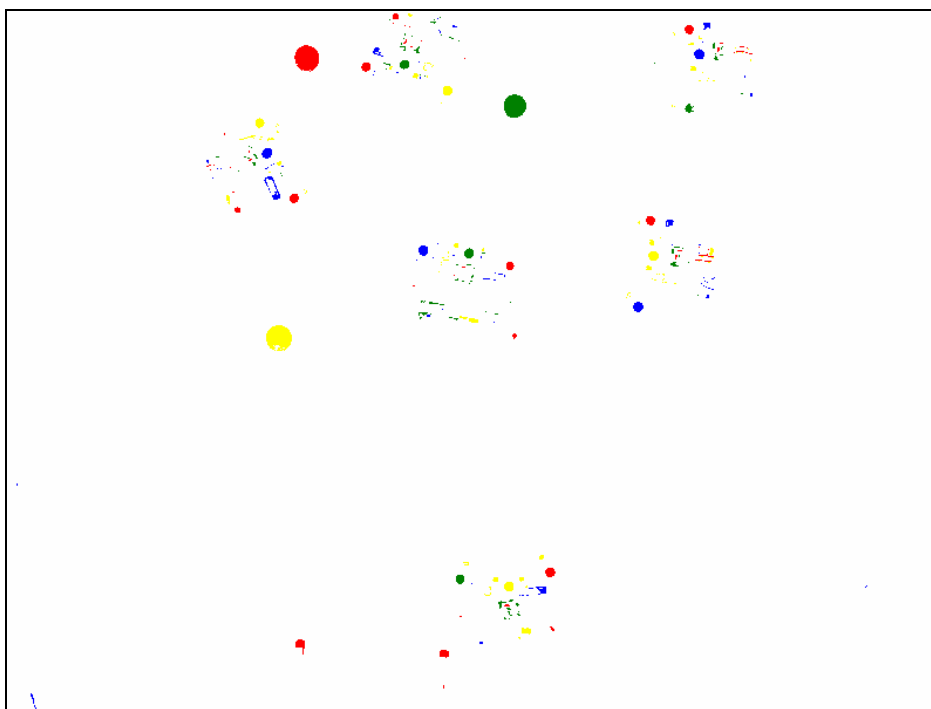
Po určení rozsahů složek pro červenou, zelenou, modrou a žlutou barvu je tedy možno přejít k vlastnímu zvýraznění těchto barev v obraze.

```

if (rh>80) and ((rh*0.5)>gh) and ((rh*0.5)>bh) then begin
    tmatrix[x,y] := clRed //cervena
end;
if (rh>80) and (gh>80) and ((rh*0.8)<gh) and ((rh*1.2)>gh) and ((rh*0.5)>bh) then begin
    tmatrix[x,y] := clYellow //zluta
end;
if (gh>42) and ((gh*0.9)>bh) and ((gh*0.5)>rh) then begin
    tmatrix[x,y] := clGreen //zelená
end;
if (bh>60) and ((bh*0.5)>rh) and ((bh*0.7)>gh) then begin
    tmatrix[x,y] := clBlue //modrá
end;

```

Rozsahy barevných složek jsou zde určeny minimální absolutní hodnotou kladné barevné složky a maximálním procentuelním podílem záporných barevných složek.



obr. 8.11: vyhledané a zvýrazněné barvy v obraze

Na obrázku 8.11 vidíme výsledek provedení zvýraznění barev ve snímané scéně. Jsou zde zvýrazněny všechny barvy, které nás zajímají z hlediska dalšího zpracování obrazu. Po zhodnocení můžeme konstatovat, že identifikační značky jsou nyní v obraze velice dobře patrné, i když ne všechny mají ideální tvar kruhových teček. V obraze je teď sice také velké množství barevných skvrn různých barev, velikostí a tvarů, ale odlišit je od identifikačních znaků by nemělo činit potíže. Toto rozlišení může být provedeno jak podle velikosti, tak podle tvaru barevné skvrny. Po podrobnějším rozboru můžeme říci, že identifikační tečky představují největší a nejkompaktnější barevné skvrny, které jsou na robotech.

V obraze jsou také velice dobře patrné míče, které má robot uchopit. Můžeme říci, že tyto míče jsou nyní velice jednoznačně a snadno identifikovatelné, nelze je zaměnit s žádným jiným objektem, ať už se týká jejich tvaru nebo velikosti, která přesahuje všechny ostatní objekty v obraze.

Co se týče robotů, tak ti jsou v tomto obraze zvýrazněných barev prezentováni shlukem skvrn různých barev, velikostí a tvarů. Netvoří sice žádný kompaktní celek, nevidíme také jejich hranici, ale přesto můžeme velice snadno odhadnout plochu, kterou zabírají.

Další objekty, které se vyskytovaly ve snímané scéně, byly části židlí a stolů. Ty nyní nejsou vůbec patrné, ve zpracovaném obraze se vůbec neukázaly. Lze říci, že je to tím, že tyto objekty nejsou nijak výrazně barevné.

Po porovnání tohoto obrazu s obrazem vzniklým prahováním si také můžeme všimnout, že v tomto obraze nám, na rozdíl od předcházejícího, netvoří žádné problémy stíny. Není jimi tedy ovlivněna například hranice robota, nebo hranice míče. To je díky naprosto odlišnému způsobu práce této metody, která vyhledává výrazné barevné oblasti, což tmavá oblast způsobená stínem nemůže být.

Tato metoda pro nalezení a zvýraznění určitých barev v obraze je tedy pro další zpracování velice užitečná.

8.4.4 Odstranění šumu, ostření obrazu

Jako další krok jsem se rozhodl vyzkoušet vliv různých filtrů na odstraňování šumu, vyhlazování a ostření obrazu.

Tyto filtry jsou založeny na značné nadbytečnosti dat v obraze. Pro výpočet nové hodnoty jasu v určitém bodě využívají vždy jistého okolí tohoto bodu. Problém s těmito metodami je, že většinou způsobí i nechtěné rozmazání obrazu. Proto jsou některé metody navrženy jako metody zachovávající hrany. Jsou většinou založeny na tom, že pro výpočet hodnoty jasu v bodě je vybráno pouze okolí s podobnými vlastnostmi jako uvažovaný bod. Všechny tyto metody fungují pouze pro odstranění drobného šumu, nejsou schopny napravit rozsáhlé poškození obrazu.

8.4.4.1 Průměrování

Průměrování je nejjednodušší metoda pro odstranění šumu. Její výsledky jsou použitelné, pokud je velikost šumu menší než nejmenší velikost objektu v obraze. Její použití ale vždy vede alespoň k mírnému rozmazání hran, což je vážná nevýhoda. Průměrování je speciální případ diskrétní konvoluce. Pro 3 x 3 sousedství má konvoluční maska tvar:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (8.1)$$

Výsledek průměrování 3 x 3 a 7 x 7 je ukázán na obrázku 8.12. Je jasně vidět, že tato metoda vede ke značnému rozmazání obrazu a zeslabení hran, její využití je tedy malé.



Obr. 8.12: Původní obraz, výsledek pro průměrování 3x3 a 7x7

8.4.4.2 Gaussův filtr

Filtrace pomocí Gaussova filtru je velice podobná průměrování, rozdíl je, že v konvoluční masce filtru je posílen význam středu.

Konvoluční maska má poté tvar:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (8.2)$$

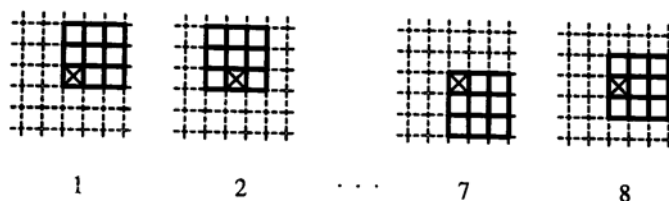
Větší konvoluční masky jsou tvořeny analogicky, podle Gaussova rozdělení. Výsledek filtrace Gaussovým filtrem je na obrázku 8.13. Po porovnání s průměrováním je vidět, že efekt rozmazání obrazu je o něco zeslaben, ale rozmazání je pořád jasně patrné.



Obr. 8.13: Původní obraz, výsledek Gaussovy filtrace 3x3 a 7x7

8.4.4.3 Průměrování s využitím rotační masky

Průměrování s využitím rotační masky je metoda, která zabraňuje rozmazání hran v obraze tím, že hledá homogenní sousedství uvažovaného bodu, které poté použije k průměrování obrazu. Pro masku velikosti 3 x 3 body máme osm různých možností volby oblasti, která pokrývá 5 x 5 sousedství daného bodu. Možnosti jsou naznačeny na obrázku 8.14.



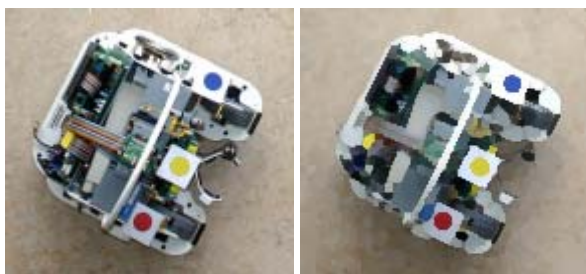
Obr. 8.14: Rotační maska

Pro volbu masky, kterou nakonec použijeme, je nejprve určena disperze všech osmi masek, a poté vybrána maska s nejmenší hodnotou disperze. Velikost disperze se spočítá podle vztahu:

$$\sigma^2 = \frac{1}{n} \left\{ \sum_{(i,j) \in R} \left[g(i,j) - \frac{1}{n} \sum_{(i,j) \in R} g(i,j) \right]^2 \right\} \quad (8.3)$$

kde n je počet bodů v oblasti R a g je vstupní obraz.

Výsledek filtrace rotační maskou je na obrázku 8.15. Je dobře patrné, nejen že nedošlo k rozmazání obrazu a zeslabení hran, ale že obraz je ve skutečnosti ještě ostřejší než originál. Šum a drobné odchylky jsou přitom dokonale odstraněny.

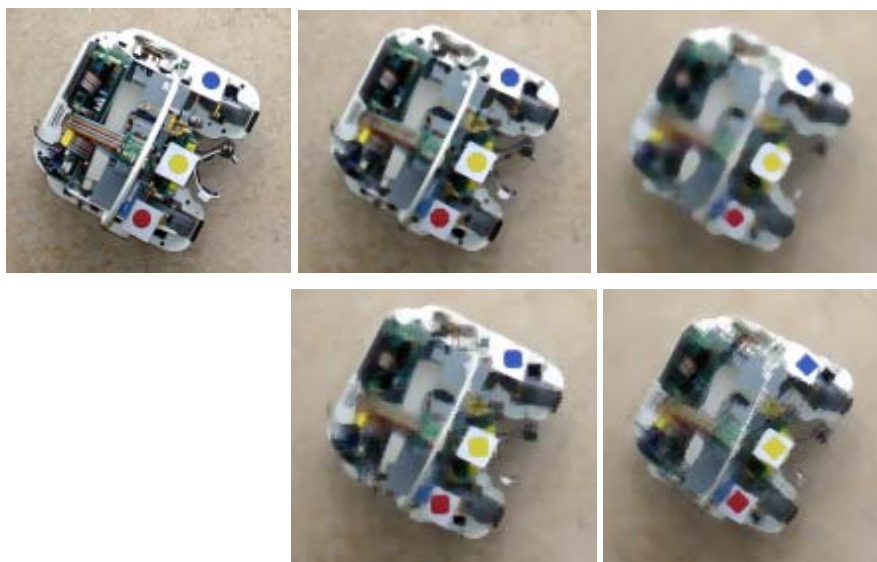


Obr. 8.15: Původní obraz, výsledek filtrace rotační maskou

8.4.4.4 Filtrace pomocí mediánu

Filtrace pomocí mediánu je nelineární vyhlazovací metoda, která omezuje rozmazání hran. Jádrem této metody je nahrazení daného bodu mediánem jeho sousedství. Uvažované sousedství zde může být nejen čtvercové, ale třeba i kříž nebo diagonální kříž.

Výsledek filtrace pomocí mediánu maskou různých velikostí a tvarů je na obrázku 8.16. Je jasně vidět že použití této metody vede k odstranění impulsního šumu a přitom dobrému zachování hran v obraze. Hlavní nevýhoda čtvercového filtru je poškození tenkých čar a ostrých rohů v obraze, tomu se lze částečně vyhnout použitím jiného tvaru masky.



Obr. 8.16: Původní obraz, výsledek filtrace pomocí mediánu 3x3, 7x7; použití křížové masky a diagonální masky 7x7

8.4.4.5 Filtrace minima, filtrace maxima

Tyto filtry jsou podobné filtru pomocí mediánu, akorát s tím rozdílem, že se daný bod nenahrazuje mediánem jeho okolí, ale buď jeho maximem nebo minimem. Filtry mohou sloužit k potlačení šumu ve světlých nebo tmavých částech obrazu.

Výsledky jsou ukázány na obrázku 8.17.



Obr. 8.17: Původní obraz, výsledek eroze 3x3; dilatace 3x3

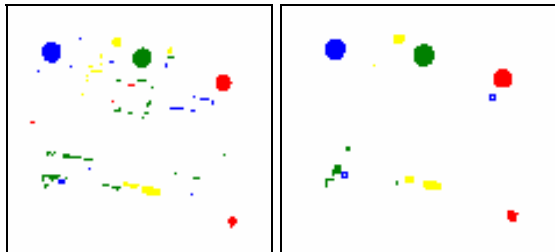
8.4.4.6 Zhodnocení výsledků

Všechny tyto metody jistým způsobem vyhlazovaly původní obraz. Můžeme je rozdělit na metody které nezachovávají hrany – průměrování a Gaussův filtr, a metody, jejichž použití vede k zachování nebo dokonce zvýraznění hran – průměrování rotační maskou, filtrace pomocí mediánu a filtrace minima, maxima.

Zvlášť zajímavé výsledky poskytuje metoda průměrování rotační maskou. Při jejím použití dojde ke značnému zvýraznění hran. Velice užitečné výsledky má použití právě této metody, v kombinaci s předchozí úpravou – se zvýrazněním barev. Pokud aplikujeme algoritmus pro zvýraznění barev na obraz filtrovaný touto metodou, pak dojde ke značné redukci nevýznamných

barevných skvrn a zároveň k zesílení identifikačních značek. Je to dáno jejich většími rozměry a kompaktnějším tvarem.

Otázkou ovšem je, jestli tuto filtraci má smysl provádět, a to především vzhledem k její časové náročnosti. Nepodstatných barevných skvrn se můžeme snadno zbavit také při další fázi zpracování.



Obr. 8.18: výsledek vyhledání barev v původním obraze, v obraze po filtraci rotační maskou

8.4.5 Detekce hran

Další velice často používanou metodou zpracování obrazu je detekce hran. Hranu v obraze vnímáme tam, kde dochází k výrazné změně sousedních pixelů. Hrana je vysokofrekvenční operace a proto je její zvýraznění inverzní operací k odstranění šumu. Hrana je určena gradientem, tzn. velikostí a směrem.

Hrany můžeme v obraze jen detekovat, nebo je přidat do původního obrazu – dochází potom k ostření obrazu. Ostření obrazu probíhá podle vztahu:

$$f(i, j) = g(i, j) - CS(i, j) \quad (8.4)$$

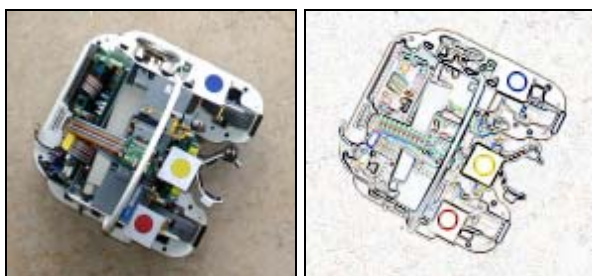
kde f je výstupní obraz, g je vstupní obraz, C je koeficient síly ostření a S je velikost gradientu v bodě.

8.4.5.1 Robertsův filtr

Je nejstarší používaný filtr. Velikost gradientu se spočítá podle vzorce:

$$|g(i, j) - g(i + 1, j + 1)| + |g(i, j + 1) - g(i + 1, j)| \quad (8.5)$$

Jeho hlavní nevýhodou je jeho vyšší citlivost k šumu, protože gradient je počítán jen z 2×2 okolí bodu.

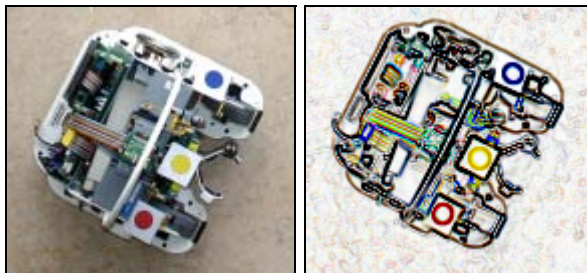


Obr. 8.19: Původní obraz, výsledek Robertsova filtru.

8.4.5.2 Sobelův filtr

Je velmi často používán jako jednoduchý detektor horizontálních a vertikálních hran. Jeho konvoluční masky jsou: (pro detekci H a V hran používány pouze masky 1 a 3)

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (8.6)$$



Obr. 8.20: Původní obraz, výsledek Sobelova filtru.

8.4.5.3 Laplaceův filtr

Velice oblíbený filtr který se často používá. Mezi jeho hlavní výhodu patří, že je všesměrový, mezi nevýhodu patří, že na hrany v obraze reaguje dvakrát.

Používají se varianty 4-sousedství, 8-sousedství, a varianta se zdůrazněným středem (není všesměrová!). Jejich masky jsou:

$$h_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad h_8 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_{s1} = \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix} \quad h_{s2} = \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \quad (8.7)$$



Obr. 8.21: Původní obraz, výsledek Laplaceova filtru (4, 8, zdůrazněný střed).

8.4.5.4 Prewittův, Robinsonův, Kirschův filtr

Tyto filtry jsou podobné Sobelovu filtru, používají však jiné konvoluční masky. Jsou také schopny detekovat hrany jen v určitém směru (horizontální, diagonální, vertikální).

Prewittův filtr:

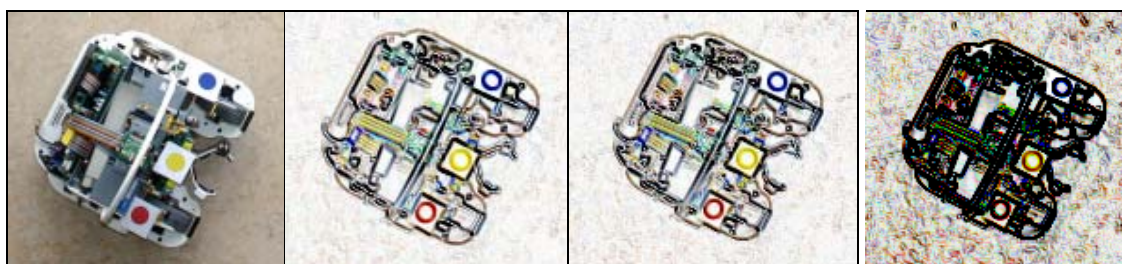
$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (8.7)$$

Robinsonův filtr

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad (8.8)$$

Kirschův filtr:

$$h_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \quad h_2 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \quad h_3 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix} \quad (8.9)$$



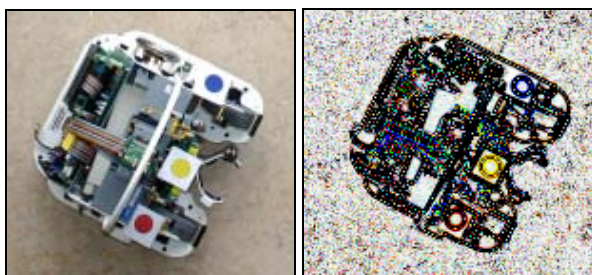
Obr. 8.22: Původní obraz, výsledek Prewittova, Robinsonova, Kirschova filtru

8.4.5.5 Inverzní LoG filtr

Je také nazýván pro svůj tvar „Mexický klobouk“. Jeho výsledky jsou obdobné Laplaceovu filtru.

Maska velikosti 5x5 má tvar:

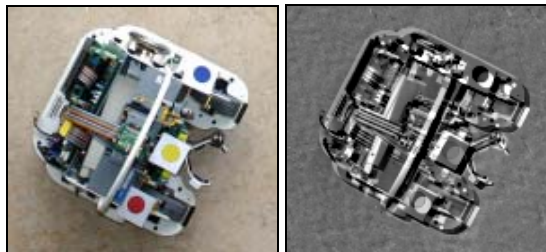
$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \quad (8.10)$$



Obr. 8.23: Původní obraz, výsledek LoG filtru

8.4.5.6 Emboss

Tento filtr, vytlačený vzor, není typický filtr pro detekci hran. Poskytuje po provedení dojem, že obraz je plastický a je jakoby vytlačen do povrchu materiálu. Je to vlastně k původnímu obrazu přičtená ještě jedna kopie, která je však posunutá o danou vzdálenost a úhel.



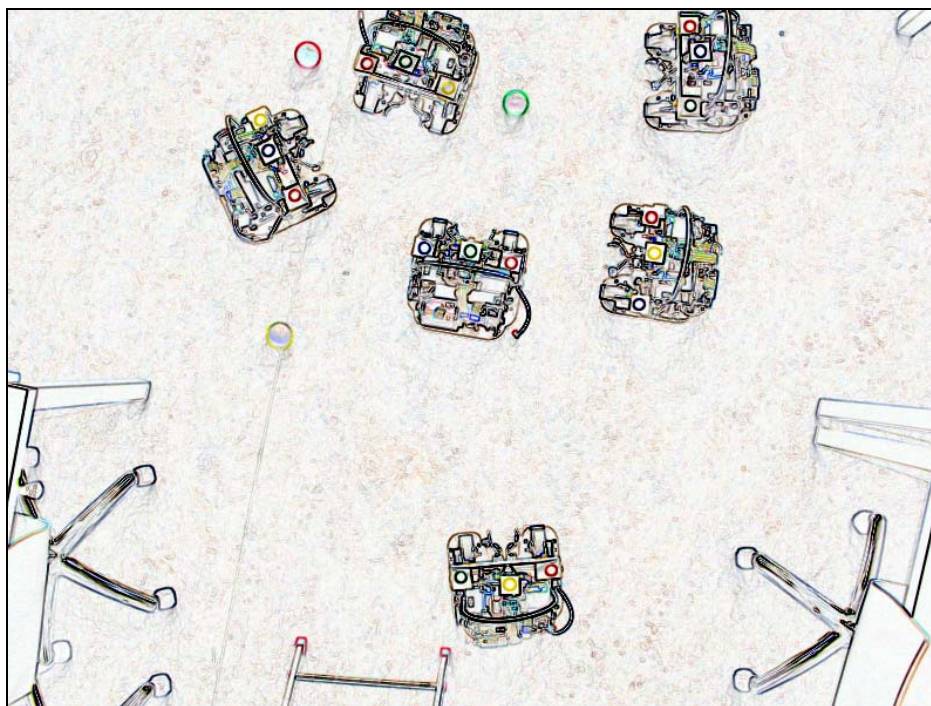
Obr. 8.24: Původní obraz, embosovaný obraz

8.4.5.7 Zhodnocení výsledků

Po prozkoumání výsledků těchto filtrů vidíme, že jsou si navzájem velmi podobné. Můžeme je rozdělit do dvou skupin. Na filtry směrové – Robertsův, Sobelův, Prewittův, Robinsonův a Kirschův, a filtry všesměrové – Laplaceův a inverzní LoG. Filtry všesměrové mají však jednu společnou nevýhodu, všechny totiž reagují na hranu v obraze dvakrát.

Nejlépe vypadají výsledky Sobelova filtru. Po zhodnocení filtrovaného obrazu celé snímané scény můžeme říci, že tento filtr spolehlivě našel hrany všech objektů. Nejlépe to jde vidět na robotech. Roboti jsou nyní v obraze velice jasně patrní, je přesně vidět celá jejich hranice.

Co se týče dalších objektů, tak ty jsou nyní také velice jasně patrné. Ani stíny vrhané těmito objekty nepůsobí žádné problémy, při detekci hran se neobjevují.



Obr. 8.25: Hrany detekované Sobelovým filtrem ve snímané scéně

8.5 Segmentace obrazu, klasifikace objektů

Segmentace obrazu a klasifikace objektů znamená, že ze snímaného obrazu dojde k vlastnímu oddělení hledaných objektů od pozadí a k jejich jednoznačné identifikaci. Pro tento účel jsou využívány výsledky předešlého předzpracování obrazu.

8.5.1 Vyhledání robotů

Prvním a asi nejdůležitějším úkolem bylo určit, kde ve snímaném obraze se nachází roboti. Po zhodnocení výsledků předzpracování jsem se rozhodl, že nejlepší metodou jak najít tyto roboty v obraze bude nějaká metoda založená na hranách obrazu.

Nerohodl jsem se však přímo využít některou ze zde popsaných metod, jejich výsledky mi posloužily jen jako inspirace.

Rozhodl jsem se zjistit, kde v obraze se nachází místa s největším gradientem, neboli kde se v obraze vyskytuje nejvíce hran, nebo ještě jinak, kde je obraz nejkontrastnější.

Pro tento úkol jsem implementoval funkci, která počítá rozdíl mezi jednotlivými barevnými složkami na oblasti 20 x 20 okolí zkoumaného bodu. Toto okolí je značně rozsáhlé, a umožňuje tedy poměrně přesné určení míst, kde se v obraze nalézají roboti. Vzhledem k velikosti tohoto okolí také nedochází k žádnému ovlivnění šumem a jinými drobnými nepřesnostmi v obraze. Díky tomu, že nás nezajímají přímo jednotlivé hrany, ale jen jestli a nakolik se v dané oblasti vyskytují, může být také tento způsob podstatně rychlejší, než metody detekce hran.

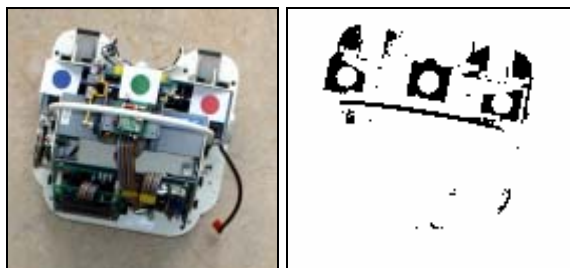
```
result:=result+abs(rmatrix[i,j]-lastr)+abs(gmatrix[i,j]-lastg)+abs(bmatrix[i,j]-lastb);  
lastr := rmatrix[i,j];  
lastg := gmatrix[i,j];  
lastb := bmatrix[i,j];
```

Tato metoda se ukázala být jako nejúčinnější, protože spolehlivě označí místa výskytu všech robotů v obraze, které je jinak velice obtížné určit jinými metodami, a přitom se nenechá zmást ostatními objekty, které se v obraze vyskytují.

Výhodou této metody tedy také je, že přesně označí celou plochu, kterou robot zabírá, dále se tedy můžeme zabývat už jen touto plochou s cílem ověřit, jestli se jedná skutečně o robota, identifikovat o kterého se jedná a jaké je jeho natočení v prostoru.

Pro zjištění jak je robot v prostoru natočen jsem nejprve používal způsob, kdy jsem zkoumal pozici držadla na robotu. Držadlo se totiž ukázalo být na všech snímcích dobře viditelné, není totiž nikdy zakryto jinými objekty.

Pro nalezení držadla se nejvhodnější ukázala být metoda prahování. Při hledání držadla, je v obraze daného robota prahovaného poměrně vysokým prahem, hledána delší, rovná, nebo mírně zahnutá linie. To je nutné z toho důvodu, že při poloze robota někde při okraji snímané scény, dochází ke zkreslení tohoto držadla, které pak má na fotografii tvar mírného oblouku. Vzhledem k tomu, že jiné rovné linie se u robota nevyskytují, nese tato metoda dobré výsledky.

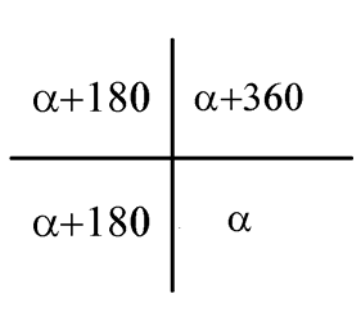


obr8.26.: originální a prahovaný obrázek robota, držadlo tvoří mírný oblouk

Po zavedení jednoznačných identifikačních značek jsem ale usoudil, že vhodnější metodou bude využít přímo těchto identifikačních značek i k zjištění orientace robota. Pokud tedy známe pozici první a poslední identifikační tečky na robotu (viz. další kapitola), můžeme orientaci robota určit jednoznačně podle vzájemné pozice těchto dvou teček. První a poslední tečka je brána z toho důvodu, že prostřední tečka, při použití druhé identifikační sady, leží výš, a její pozice ve 2D obraze tedy může být díky perspektivnímu zkreslení mírně posunutá. Úhel robota je pak dán vztahem:

$$\tan \alpha = \frac{y_2 - y_1}{x_2 - x_1} \quad (8.11)$$

kde $[x_1, y_1]$ jsou souřadnice první tečky a $[x_2, y_2]$ jsou souřadnice poslední identifikační tečky. Musíme pak akorát ještě vzít do úvahy, v jakém kvadrantu se druhá tečka oproti tečce první vyskytuje, a podle toho pak velikost úhlu případně upravit přičtením 180 nebo 360 stupňů.



obr8.27.: úhel natočení robota

Tímto způsobem tedy získáme velice snadno přesnou orientaci robota v prostoru. Pokud je natočen přímo směrem dolů, pak je tato velikost úhlu rovna nule, poté velikost úhlu roste po směru hodinových ručiček.

8.5.2 Vyhledání identifikačních značek

Pro vyhledání identifikačních značek jsem se rozhodl využít obraz získaný zvýrazněním barev v původním obraze. V tomto obraze jsou totiž identifikační tečky velice dobře patrné.

Můj postup byl tedy takový, že jsem prozkoumal celý obraz, který vznikl zvýrazněním barev, a pro každou nalezenou barevnou skvrnu jsem vytvořil záznam v tabulce, který obsahoval její pozici, velikost, a některé další parametry.

Barevné skvrny jsem v této tabulce také rozdělil do několika úrovní, podle jejich vlastností, nakolik je pravděpodobné, že se jedná o identifikační tečku a ne o něco jiného. Hlavním kritériem, podle kterých jsem je dělil, byla tedy jejich velikost. U identifikační tečky známe přesně její velikost a jakákoliv odchylka směrem dolů nebo nahoru tedy snižuje pravděpodobnost, že se jedná právě o identifikační značku. Dalším kritériem byla také jejich kompaktnost a tvar, tzn. nakolik se jedná o kruhovou tečku a ne o něco jiného.

Když tedy máme nalezené všechny tyto identifikační značky, můžeme podle nich identifikovat jednotlivé roboty nalezené v předešlém kroku. Při identifikaci robotů musíme rozlišovat dva případy, a to podle toho, jak jsou znaky na jednotlivé roboty umístěny a tedy která identifikační sada byla pro označení použita.

Při použití první sady je situace o něco jednodušší. Na obrázku 8.28 na první pohled vidíme, že všechny tři identifikační tečky jsou ve stejné výšce a tedy i v jedné přímce, odchylka od této přímky může být maximálně v řádu několika bodů. Také vzdálenost mezi první, druhou a třetí tečkou musí být vždy stejná, tato vzdálenost se může lišit také maximálně jen o několik bodů. Nedochází k žádným větším odchylkám způsobeným perspektivním skreslením.

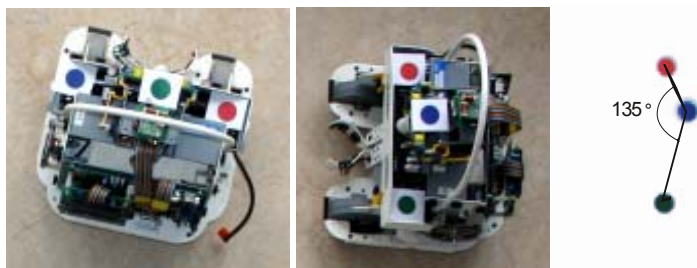


obr. 8.28: robot označený první sadou

Při použití druhé sady, je situace o něco málo složitější. Zde totiž leží prostřední znak o něco výše než dva krajní, což může působit právě díky perspektivnímu zkreslení jisté odchylky. Situaci dobře ilustruje obrázek 8.29. Můžeme zde vidět, že vzdálenost od prostředního znaku ke dvěma krajním nemusí být ve všech případech stejná, tato odchylka může být poměrně významná. Také nemůžeme říci, že všechny tři tečky musí tvořit jednu přímku, úhly mezi první a druhou a druhou a třetí tečkou mohou být značně rozdílné. Naproti tomu dva krajní znaky jsou ve stejné výšce, vzájemné zkreslení je tedy minimální. Díky tomu můžeme poměrně přesně říci, jaká vzdálenost mezi těmito znaky musí být, a také, že jejich vzájemný úhel odpovídá úhlu orientace celého robota.

Všechny tyto poznatky musíme tedy vzít do úvahy když tvoříme pravidla, které identifikují tyto roboty podle identifikačních značek. Podle těchto pravidel je potom nutné rozhodnout, zda objekt, který byl nalezen v předešlém kroku při hledání kontrastních objektů, je skutečně jeden z robotů, a případně tohoto robota jednoznačně identifikovat.

Zvláštní situace ještě nastává, když je některá z identifikačních teček zakryta, případně robot není v záběru vidět celý. V takovém případě je nutné nejprve identifikovat všechny ostatní roboty, kteří jsou v záběru kompletní, a poté se pokusit nalézt chybějícího robota mezi nepřirazenými objekty vzniklými v předešlém kroku, identifikovat jej na základě dvou zbylých identifikačních značek. Protože se tento případ většinou týká jen konkrétního robota, o kterém víme, kde by měl všechny znaky mít, tak je značná šance, že tato identifikace na základě dvou značek bude úspěšná.



obr. 8.29: roboti označení druhou sadou: první robot je přibližně ve středu záběru, odchylky nejsou významné; druhý robot je v rohu záběru, odchylky způsobené perspektivním zkreslením jsou značné; pokud by všechny body byly v přímce, úhel by byl 180 stupňů, zde je však jen 135 stupňů

8.5.3 Vyhledání míčů

Jako poslední úkol bylo vyhledání jednotlivých míčů které má robot uchopit. Tento úkol je nyní už poměrně jednoduchý. Pokud se podíváme na obrázek 8.11, tak vidíme, že tyto míče jsou v obraze, kde byly zvýrazněny barvy opravdu jedinečné a nepřehlédnutelné, ať už se týká tvaru nebo velikosti.

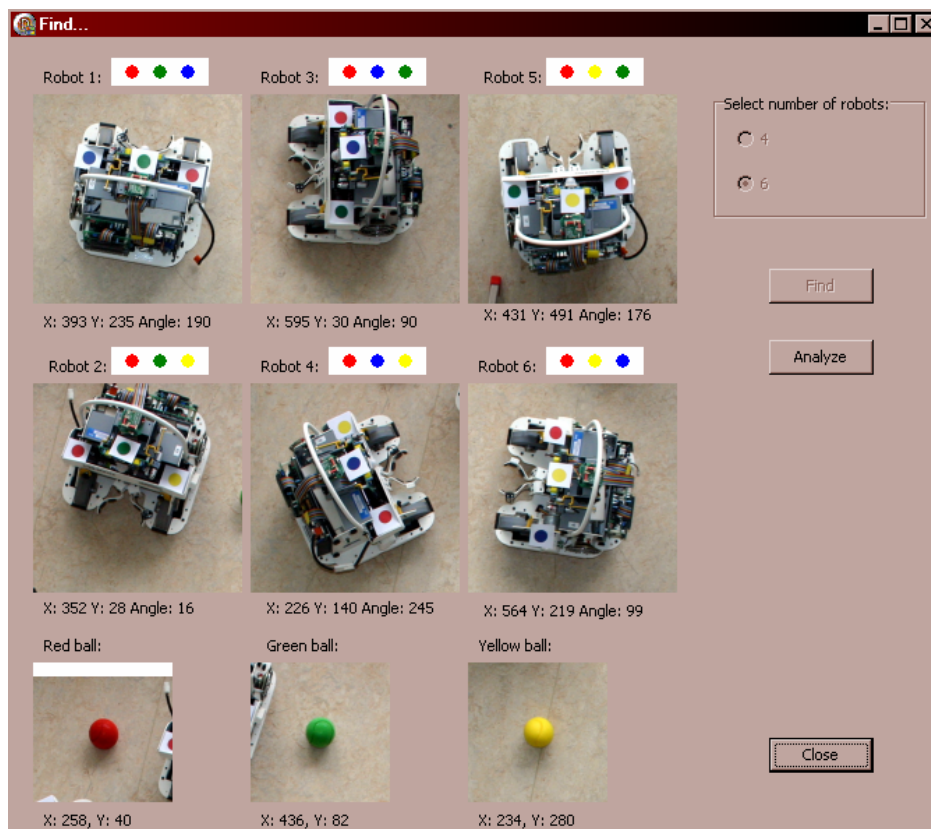
Jejich vyhledání tedy můžeme provést zároveň s předešlým krokem, kdy vyhledáváme identifikační značky v obraze. Stačí do pravidel, podle kterých klasifikujeme a zařazujeme do tabulky jednotlivé barevné skvrny, přidat pravidlo týkající se těchto větších míčů. Známe jejich velikost, tvar, víme i to, že jsou to velice kompaktní útvary. Vzhledem k tomu, že se jedná o kulaté míče, tak se nemusíme zabývat ani jejich perspektivním zkreslením.

Po provedení tohoto vyhledání identifikačních značek, máme tedy už přímo k dispozici i polohy těchto míčů, není potřeba žádné další zpracování.

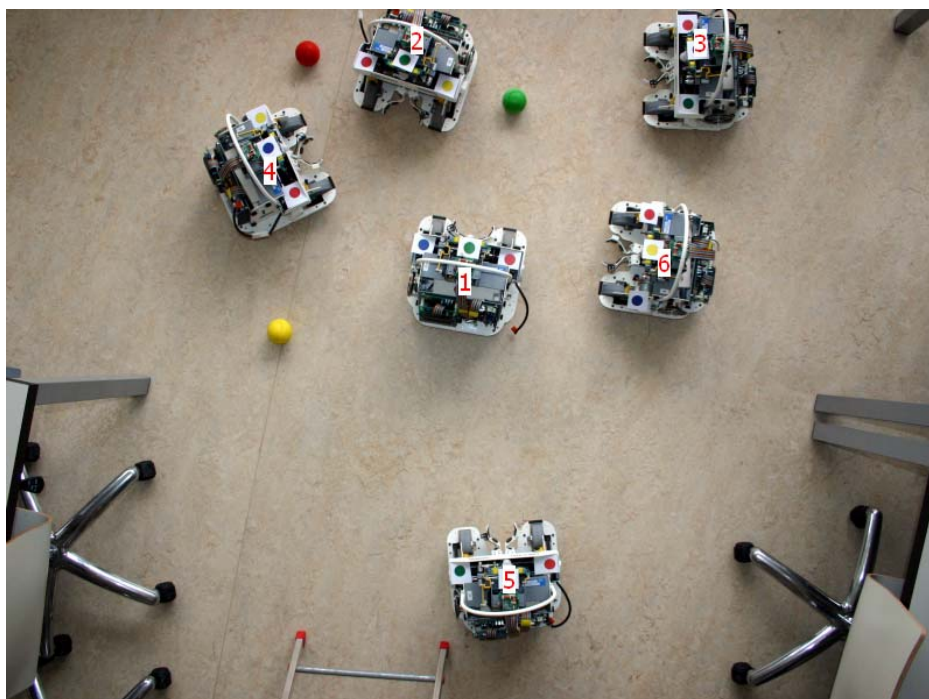
Na obrázku 8.30 máme celkový pohled na obrazovku programu po provedení vyhledávání všech objektů. V dolní části okna vidíme jednotlivé barevné míče, pod každým jsou jeho souřadnice. Je nutné připomenout, že souřadnice mají počátek v levém horním rohu vstupního obrazu.

Hlavní část tohoto okna zabírají jednotliví nalezení roboti, nad každým je zobrazeno jeho číselné a grafické označení. Pod nimi jsou zobrazeny jejich souřadnice a úhel natočení každého z nich.

Obrázek 8.31 potom ukazuje pohled na vstupní obraz po provedeném vyhledávání. Zde je každý robot nyní označen svým identifikačním číslem.



obr. 8.30: vyhledání všech objektů



obr. 8.31: označení robotů ve vstupním obraze

8.6 Vztahy mezi objekty v obraze

Poslední úkol který jsem řešil, bylo zjistit jaké panují vztahy mezi jednotlivými objekty v obraze. To například znamená zjistit, jaká je vzdálenost mezi každým robotem a každým míčem, a také to, zda má robot k míči volnou cestu, nebo zda je nějak blokován.

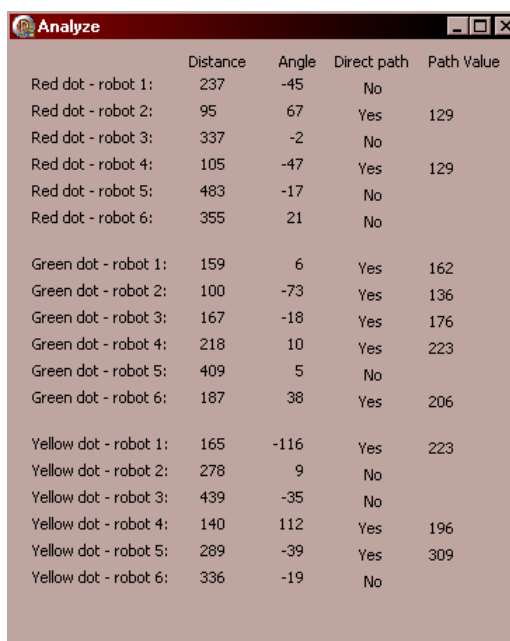
Jelikož už známe polohu všech objektů, byl tento úkol poměrně jednoduchý. Vzdálenost robota k míči se vypočítá podle vztahu 3.4. Zde je nutno podotknout, že se uvažuje vzdálenost od těžiště robota k těžišti míče. Pokud bychom chtěli znát kontaktní vzdálenost, byl by tento údaj menší.

Další úkol byl zjistit, o kolik stupňů by se musel robot otočit, kdyby se chtěl nasměrovat na míč. Pro provedení tohoto úkolu bylo tedy nutné znát aktuální úhel natočení robota a úhel mezi robotem a míčem. Oba dva tyto údaje jsou známy, takže tento úhel lze snadno spočítat.

Tento úhel je pak udáván jako číslo v rozsahu -180 až +180 stupňů. Kladný směr znamená otočení po směru hodinových ručiček, záporný směr potom otočení robota proti směru hodinových ručiček.

Jako další provádím zjištění, zda má robot k cíli volnou cestu. Zde zjišťuji, zda robot může rovnou jet a uchopit míč, nebo zda mu v tom brání jiný robot nebo jestli je v cestě nějaký jiný míč. Vzhledem k tomu, že známe už pozice všech těchto objektů, tak můžeme lehce prozkoumat, zda některý z těchto objektů se nachází ve stejném směru jako míč, a zda mu tedy blokuje cestu.

Dále ještě zjišťuji cenu cesty, to znamená, jaká doba by asi byla nutná k tomu, aby se robot dostal až k míči. Zde beru do úvahy jak údaj o kolik stupňů se musí robot natočit, tak vzdálenost mezi robotem a cílem. Vycházím z toho, že otočení robota o jeden stupeň trvá půl časové jednotky, a posun robota o jeden bod trvá jednu časovou jednotku. Z tohoto údaje tedy lze určit, který z robotů má nejvýhodnější pozici, aby se vydal uchopit míč.



	Distance	Angle	Direct path	Path Value
Red dot - robot 1:	237	-45	No	
Red dot - robot 2:	95	67	Yes	129
Red dot - robot 3:	337	-2	No	
Red dot - robot 4:	105	-47	Yes	129
Red dot - robot 5:	483	-17	No	
Red dot - robot 6:	355	21	No	
Green dot - robot 1:	159	6	Yes	162
Green dot - robot 2:	100	-73	Yes	136
Green dot - robot 3:	167	-18	Yes	176
Green dot - robot 4:	218	10	Yes	223
Green dot - robot 5:	409	5	No	
Green dot - robot 6:	187	38	Yes	206
Yellow dot - robot 1:	165	-116	Yes	223
Yellow dot - robot 2:	278	9	No	
Yellow dot - robot 3:	439	-35	No	
Yellow dot - robot 4:	140	112	Yes	196
Yellow dot - robot 5:	289	-39	Yes	309
Yellow dot - robot 6:	336	-19	No	

obr. 8.32:vztahy mezi objekty

9 Závěr

Na základě požadavků jsem navrhl program pro lokalizaci robotů Trilobot a objektu který má robot uchopit v laboratoři. Pro lokalizaci a identifikaci jednotlivých robotů lze s výhodou použít různé značky, z nichž nejvýhodnější se ukázalo být označit každého robota jinou sadou různě barevných identifikačních teček.

Tento navržený program dosahuje poměrně spolehlivých a přesných výsledků. Ve všech vstupních obrazech dokáže jednoznačně určit polohu všech robotů, identifikovat je, a to i pokud je identifikační znak zakryt či pokud je robot v obraze jen zčásti. Také orientaci robota je tento navržený program schopen velice přesně a spolehlivě určit.

Co se týče objektů které má robot uchopit, tak jako tyto objekty byly vybrány tři různé barevné míče. Také při určování poloh těchto objektů je dosaženo velice přesných výsledků.

Program je také schopen zjistit, jaké jsou vzájemné vztahy mezi všemi těmito objekty, dokáže dodat informace potřebné k navedení robota k míči.

Při řešení tohoto projektu jsem se seznámil se značným množstvím různých metod a postupů používaných ke zpracování a identifikaci obrazu. Některé z nich jsem později i využil v tomto projektu, dalšími jsem se inspiroval při navrhování vlastních metod, které jsem potřeboval.

Směr dalšího vývoje tohoto projektu by se mohl ubírat například ke zpracování obrazu snímaného digitální kamerou v reálném čase. Digitální kamera, nebo případně digitální fotoaparát by byl připevněn ve stacionární pozici u stropu nad celou scénou. Toto by pak přineslo další možnosti, například aby pozice jednotlivých robotů jim byla bezdrátově předávána, roboti by pak také mohli být přímo navigováni k nějakému objektu, přičemž vzájemné pozice všech robotů by byly neustále kontrolovány a analyzovány tímto programem.

Co mě při zpracování tohoto projektu velice zaujalo, bylo obrovské množství různých metod úprav a zpracování digitálního obrazu. Některé z těchto metod doslova vytěží a zvýrazní některé informace, které se vyskytují v obraze, ale které jsou jinak při běžném zkoumání pro člověka jen obtížně postřehnutelné. Jiné z těchto metod jsou zase schopny opravit i rozsáhlejší poškození obrazu, bez toho aby obraz nějakým způsobem utrpěl.

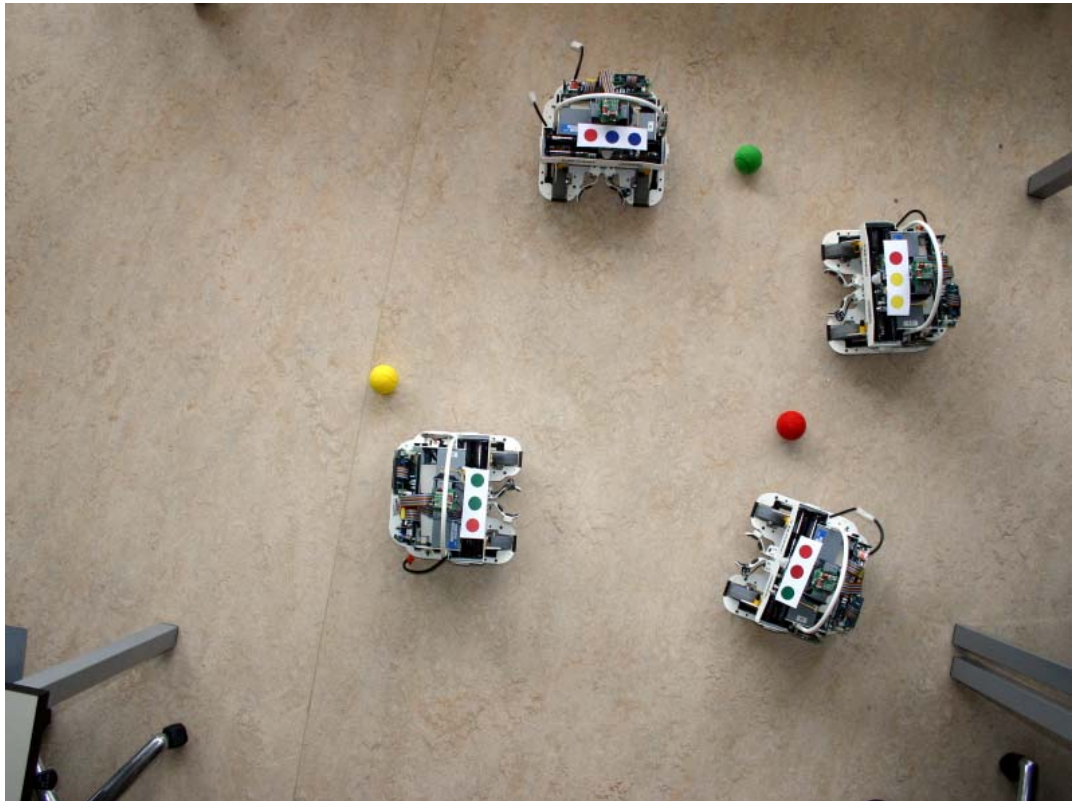
Literatura

- [1] Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision
Brooks/Cole publishing Company 1999
ISBN 0-534-95393-X
- [2] Žára, J., Beneš, B., Sochor, J., Felkel, P.: Moderní počítačová grafika
Computer Press 2004
ISBN 80-251-0454-0
- [3] Lischner R.: Delphi v kostce, pohotová referenční příručka
Computer Press 2000
ISBN 80-7226-361-7
- [4] Šonka, M., Hlaváč: Počítačové vidění
Grada 1992
ISBN 80-85424-67-3

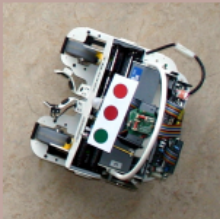
Seznam příloh

Příloha 1. CD se zdrojovými kódy programu

Příloha 2. Zdrojové obrazy a jejich výsledky po zpracování

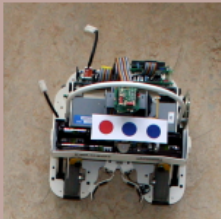


Robot 1: ● ● ●



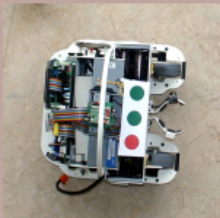
X: 593 Y: 424 Angle: 114

Robot 3: ● ● ●



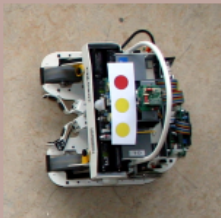
X: 443 Y: 89 Angle: 5

Robot 2: ● ● ●



X: 337 Y: 370 Angle: 277

Robot 4: ● ● ●



X: 660 Y: 212 Angle: 90

Select number of robots:


☐ 4
 ☐ 6

Find

Analyze


Close

Red ball:




X: 588, Y: 316

Green ball:

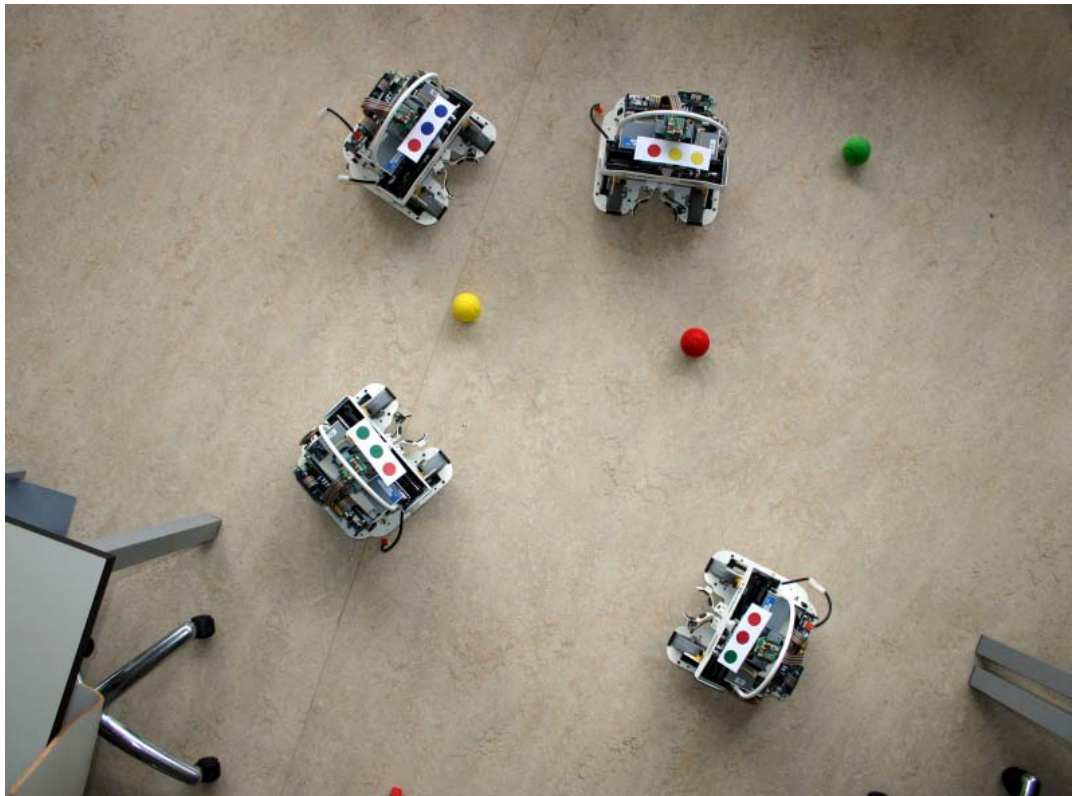


X: 556, Y: 117

Yellow ball:



X: 284, Y: 282



Robot 1:

X: 555 Y: 466 Angle: 122

Robot 3:

X: 302 Y: 111 Angle: 308

Robot 2:

X: 281 Y: 349 Angle: 234

Robot 4:

X: 490 Y: 118 Angle: 10

Select number of robots:

☒ 4
☐ 6

Find

Analyze

Red ball:

X: 516, Y: 256

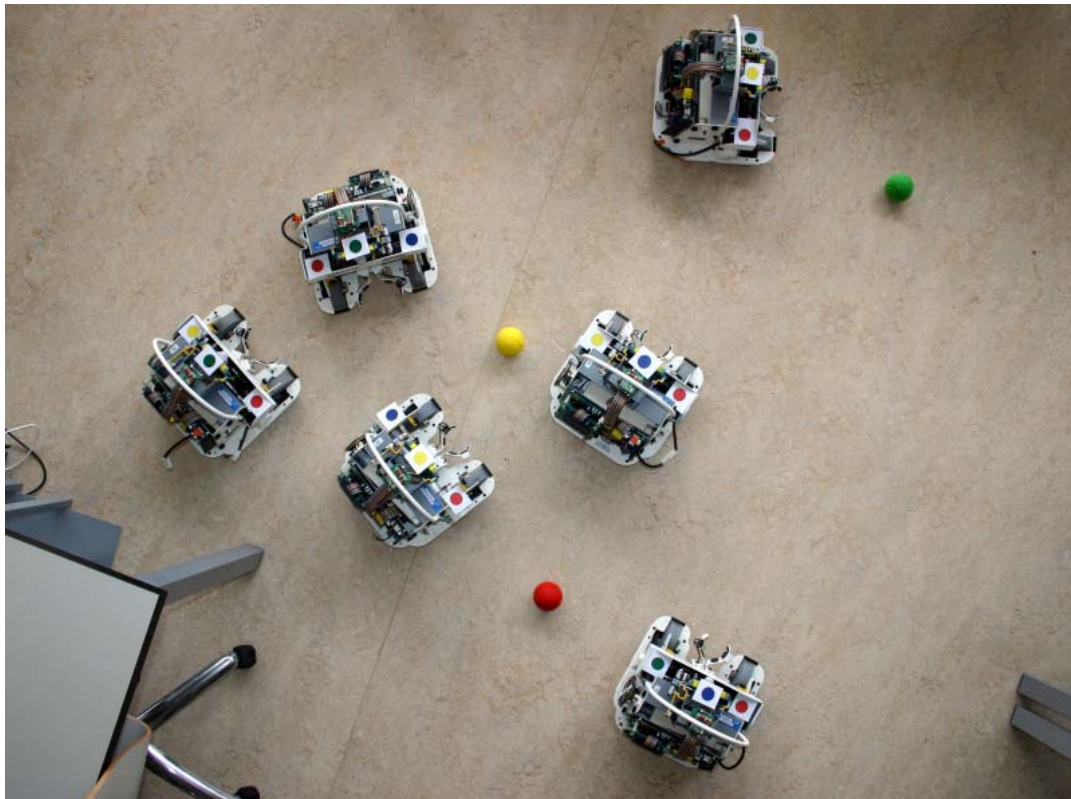
Green ball:

X: 636, Y: 112

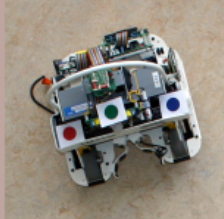
Yellow ball:

X: 346, Y: 229

Close




Robot 1: ● ● ●



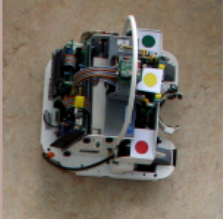
X: 266 Y: 182 Angle: 344

Robot 3: ● ● ●



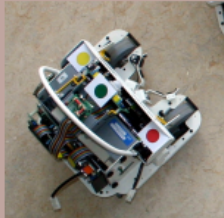
X: 510 Y: 517 Angle: 207

Robot 5: ● ● ●



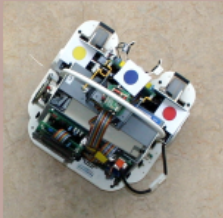
X: 534 Y: 68 Angle: 273

Robot 2: ● ● ●



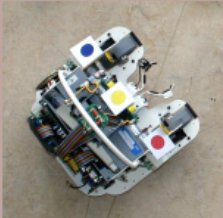
X: 164 Y: 283 Angle: 228

Robot 4: ● ● ●




X: 467 Y: 293 Angle: 215


Robot 6: ● ● ●




X: 308 Y: 351 Angle: 233

Red ball:


X: 406, Y: 444

Green ball:


X: 668, Y: 140

Yellow ball:


X: 378, Y: 254

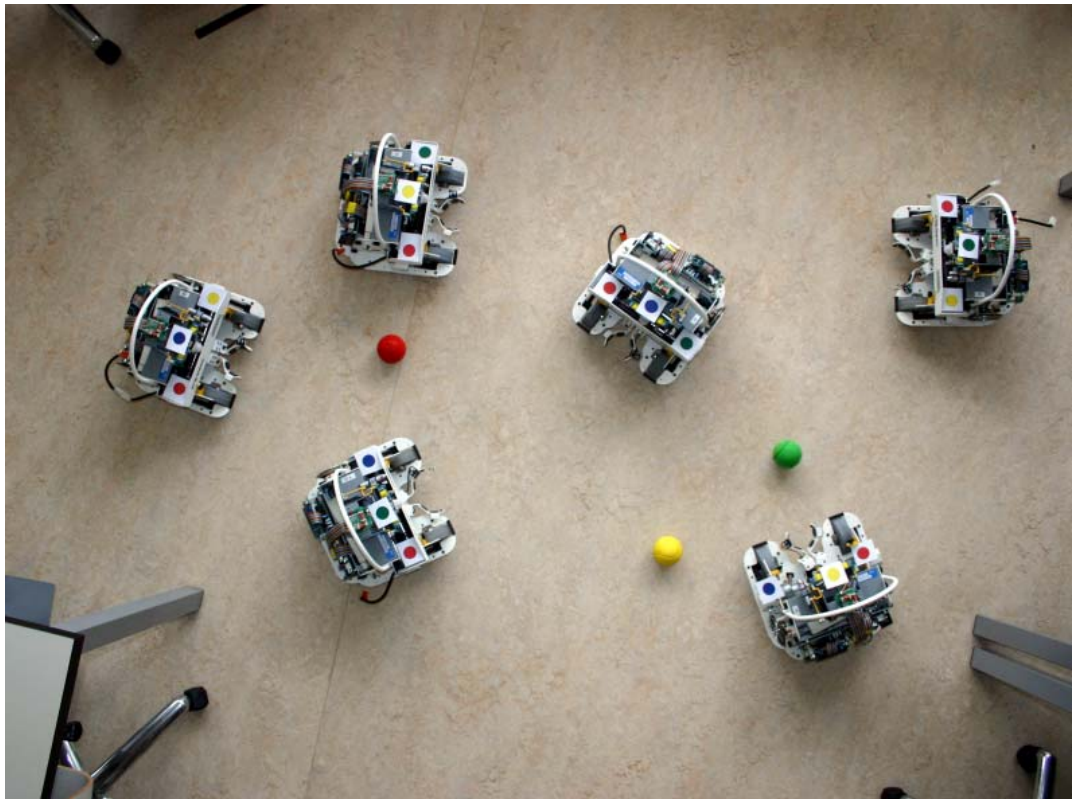
Select number of robots:

☐ 4
☒ 6

Find

Analyze

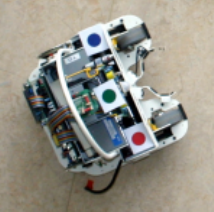
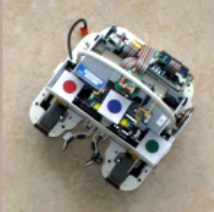
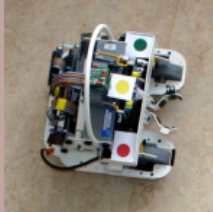
Close



Robot 1: ● ● ●

Robot 3: ● ● ●

Robot 5: ● ● ●

X: 282 Y: 388 Angle: 246

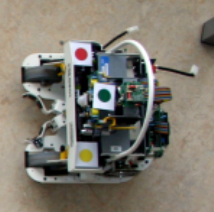
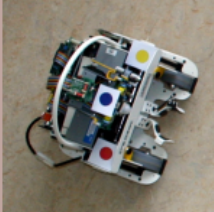
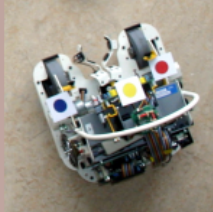
X: 480 Y: 227 Angle: 37

X: 295 Y: 155 Angle: 279

Robot 2: ● ● ●

Robot 4: ● ● ●

Robot 6: ● ● ●

X: 721 Y: 191 Angle: 87




X: 135 Y: 258 Angle: 290

X: 606 Y: 440 Angle: 160

Red ball:

Green ball:

Yellow ball:

X: 289, Y: 260

X: 584, Y: 338

X: 494, Y: 410

Select number of robots:

☐ 4
 ☒ 6

Find

Analyze

Close